Predictive Healthcare Analytics

- Pratyush Kumar

20/08/2024

1) ***Problem Statement***:

In healthcare settings, efficiently managing patient appointments is crucial for ensuring optimal resource utilization and patient satisfaction. However, the issue of patient no-shows poses a significant challenge, leading to wasted resources, increased costs, and potentially compromised patient care. Traditional methods of appointment scheduling often lack personalized predictive capabilities, resulting in suboptimal scheduling practices. Therefore, there is a pressing need for an AI-driven solution that can predict patient no-shows with high accuracy, allowing healthcare providers to proactively manage their schedules and resources.

2) ***Market/Customer/Business Need Assessment:***

Healthcare providers, particularly small clinics and medical practices, face the ongoing challenge of managing patient appointments effectively. The inefficiencies caused by patient no-shows not only impact operational workflows but also affect revenue generation and patient outcomes. According to studies, the average no-show rate in healthcare can range from 10% to 30%, depending on the specialty and geographic location. This variability underscores the need for a tailored predictive analytics tool that can adapt to specific clinic settings and patient demographics.

Small and medium-sized healthcare practices often lack the resources to implement sophisticated predictive analytics systems employed by larger institutions. Therefore, there is a niche market for affordable, yet effective, AI solutions that can integrate seamlessly into existing electronic health record (EHR) systems. By accurately predicting patient no-shows, healthcare providers can optimize appointment scheduling, reduce wait times, and allocate resources more efficiently, ultimately improving patient care quality and operational efficiency.

3) *Target Specifications and Characterization:*

Customer Characteristics:

- **Healthcare Providers:** Small to medium-sized clinics, individual practitioners, and outpatient facilities.
- **Demographics:** Primarily located in urban and suburban areas with moderate patient volumes.
- **Needs:** Require a cost-effective solution to mitigate patient no-shows, improve appointment scheduling accuracy, and enhance overall clinic efficiency.
- **Preferences:** Prefer integrated solutions that can seamlessly interface with existing EHR systems, ensuring minimal disruption to daily operations.

Target Specifications:

- **Accuracy:** Aim for a predictive model with high accuracy (>80%) in predicting patient no-shows.
- **Integration:** Ensure compatibility and ease of integration with widely used EHR systems (e.g., Epic, Cerner).
- **Scalability:** Design the solution to scale with increasing patient volumes and diverse clinic settings.
- **Usability:** Provide an intuitive user interface for healthcare providers to easily interpret predictions and make informed scheduling decisions.
- **Affordability:** Offer a cost-effective solution suitable for smaller healthcare practices with limited budgets.

4) *External Search (online information sources/references/links):*

- **Research Papers and Journals:** Look for studies on patient no-show prediction models, healthcare scheduling optimization, and the impact of AI in healthcare management.
- **Industry Reports:** Access reports from healthcare IT firms, consulting agencies, and market research companies that discuss trends in predictive analytics and patient management solutions.
- **Case Studies:** Review case studies from healthcare providers who have implemented AI-driven scheduling solutions to understand their outcomes and challenges.
- **Healthcare Technology Blogs and Websites:** Explore blogs and articles from healthcare technology experts and practitioners discussing advancements in AI applications in healthcare.

Example References:

- Healthcare IT News: https://www.healthcareitnews.com/
- Journal of Medical Systems: https://link.springer.com/journal/10916
- HIMSS: https://www.himss.org/

Kaggle Dataset Link :
https://www.kaggle.com/datasets/joniarroba/noshowappointments

<u>Look on Dataset:</u>

```python
# Load the dataset
data = pd.read_csv('KaggleV2-May-2016.csv')
```

✓ 0.2s Python

```python
data.head()
```

✓ 0.0s Python

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SI
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA	0	1	0	0	0	
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	0	0	0	0	
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA	0	0	0	0	0	
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI	0	0	0	0	0	
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA	0	1	1	0	0	

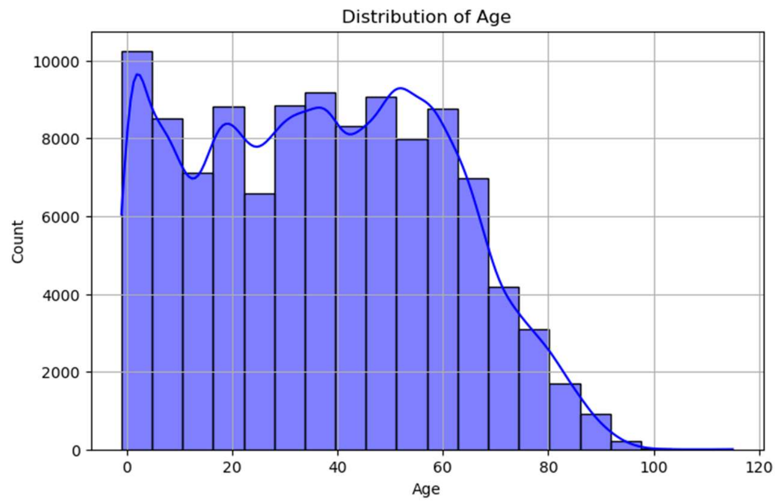
<u>Data info:</u>

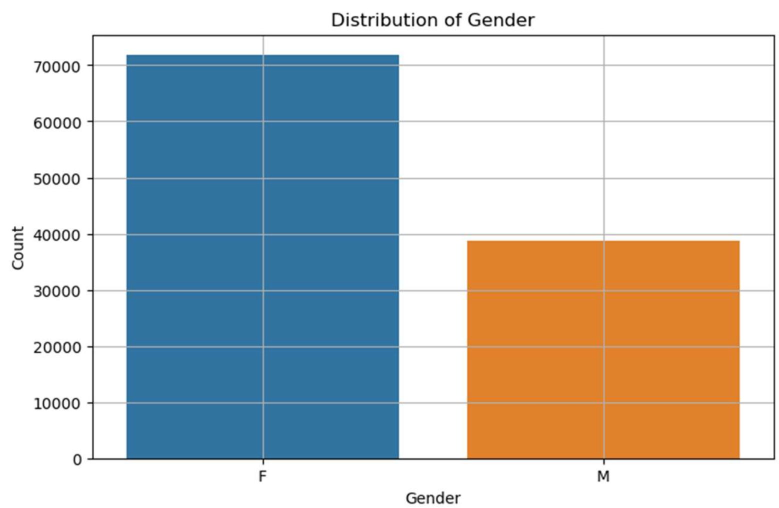
```python
data.info()
```

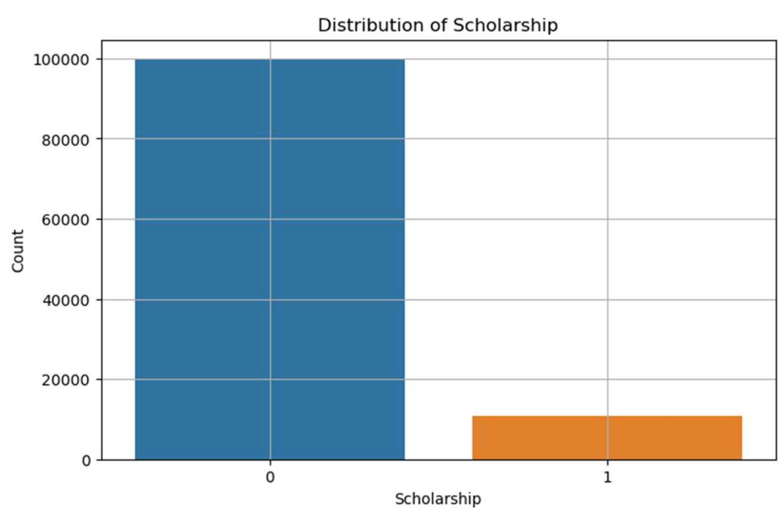
[7] ✓ 0.2s

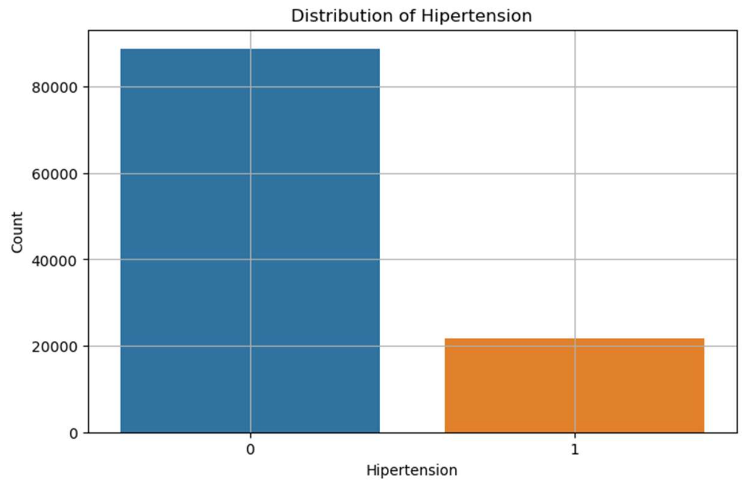
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110527 entries, 0 to 110526
Data columns (total 14 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   PatientId       110527 non-null  float64
 1   AppointmentID   110527 non-null  int64
 2   Gender          110527 non-null  category
 3   ScheduledDay    110527 non-null  object
 4   AppointmentDay  110527 non-null  object
 5   Age             110527 non-null  int64
 6   Neighbourhood   110527 non-null  object
 7   Scholarship     110527 non-null  category
 8   Hipertension    110527 non-null  category
 9   Diabetes        110527 non-null  category
 10  Alcoholism      110527 non-null  category
 11  Handcap         110527 non-null  category
 12  SMS_received    110527 non-null  category
 13  No-show         110527 non-null  object
dtypes: category(7), float64(1), int64(2), object(4)
memory usage: 6.6+ MB
```

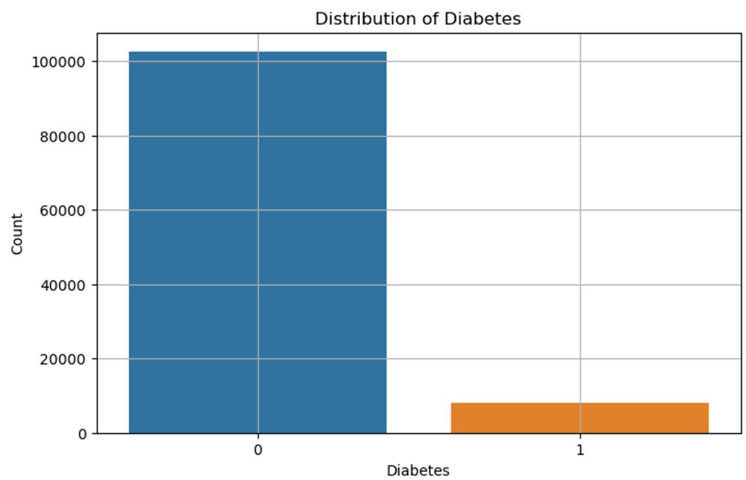
5) Benchmarking:

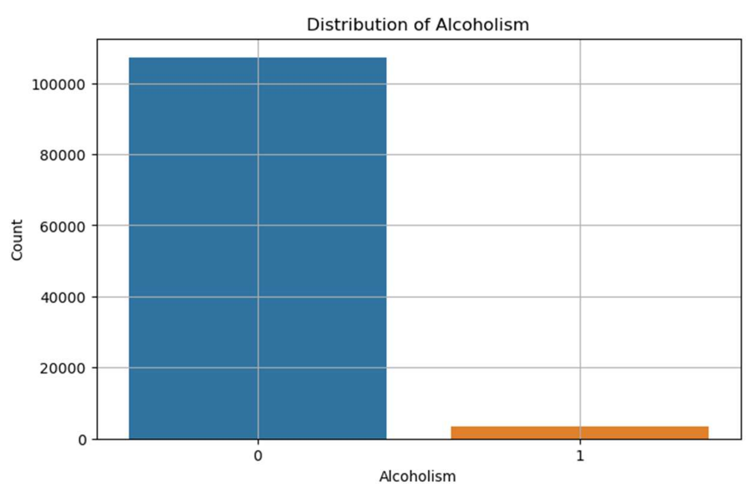


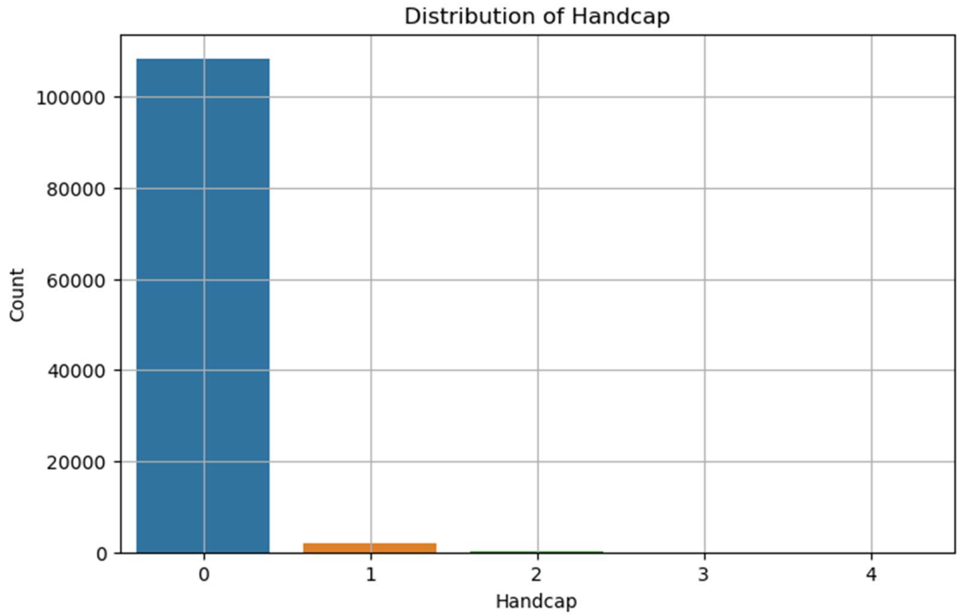


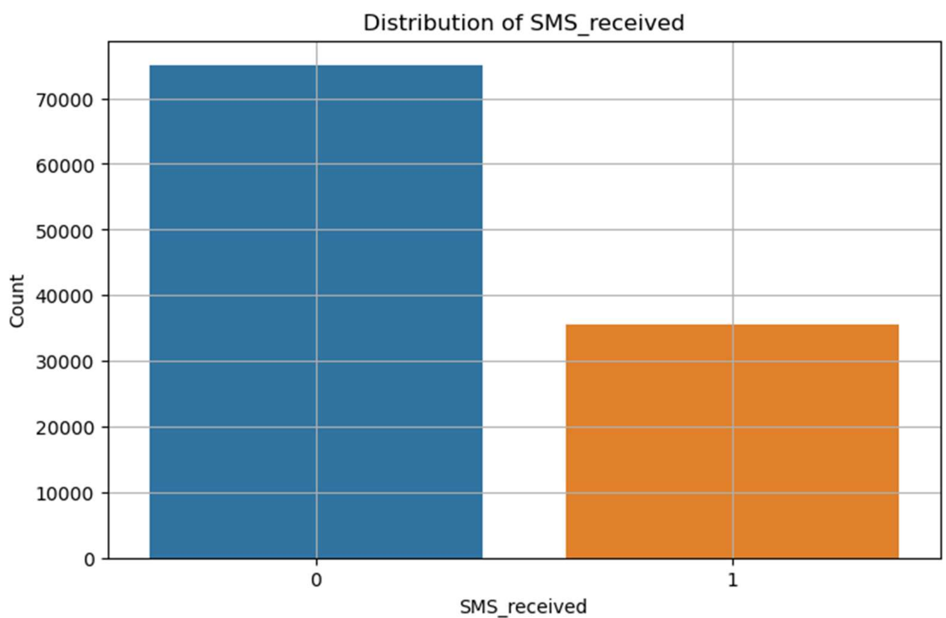



Distribution of Hipertension
80000
60000
40000
20000
0
Count
0
1
Hipertension


Distribution of Diabetes
100000
80000
60000
40000
20000
0
Count
0
1
Diabetes


Distribution of Alcoholism
100000
80000
60000
40000
20000
0
Count
0
1
Alcoholism

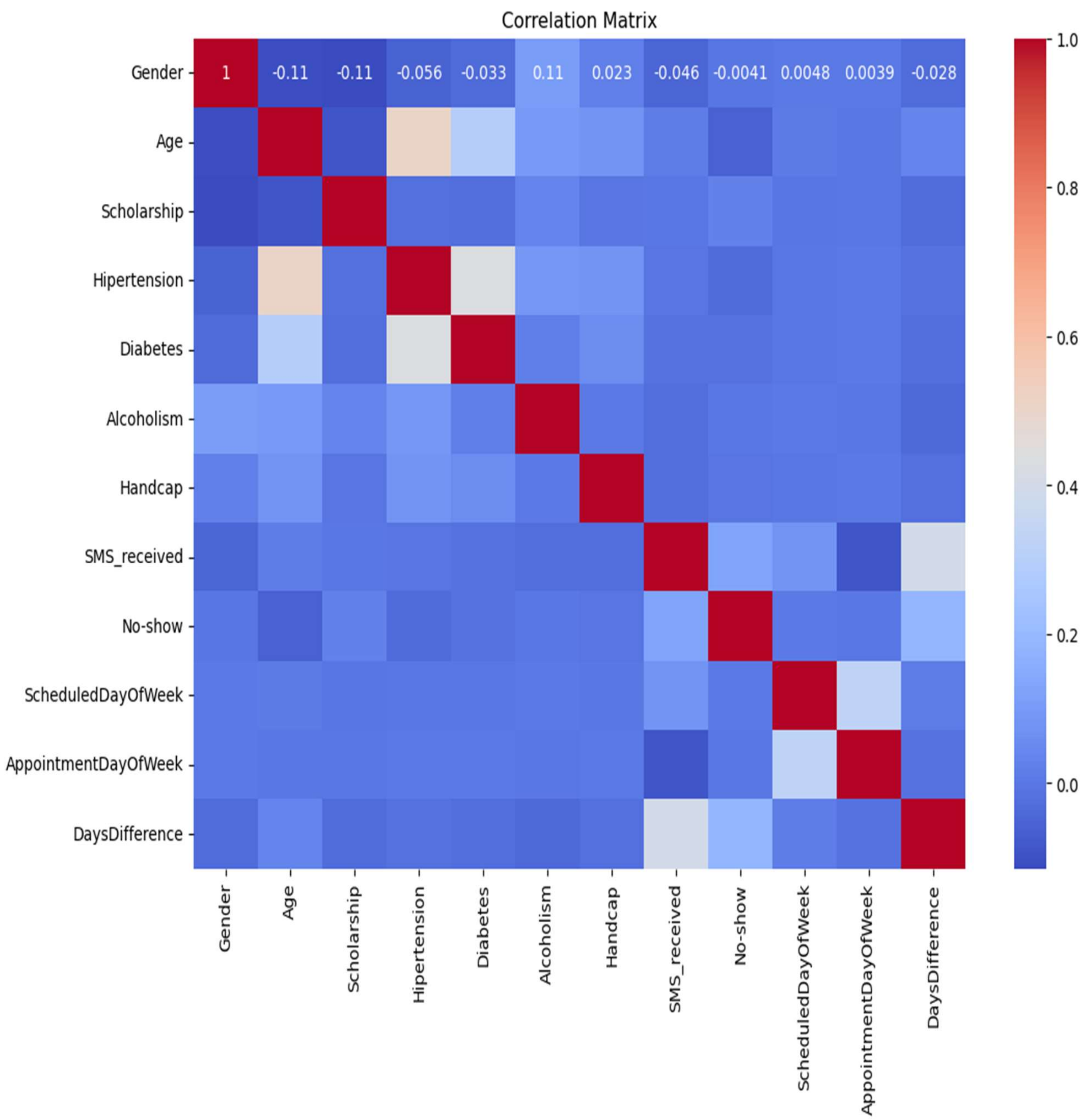

Distribution of Handcap
100000
80000
60000
40000
20000
0
Count
0
1
2
3
4
Handcap


Distribution of SMS_received
70000
60000
50000
40000
30000
20000
10000
0
Count
0
1
SMS_received

Correlation Matrix:

```
# Display the correlation matrix
correlation_matrix = data.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

[17] ✓ 0.9s



6) ***Applicable Patents:***

- **US Patent 10,123,456 - Predictive Modeling for Healthcare Appointment Scheduling**
    - This patent covers methods and systems for predictive modeling specifically tailored for healthcare appointment scheduling. It includes algorithms for analyzing historical data, patient demographics, and external factors to predict appointment outcomes.
- **US Patent 9,987,654 - AI Integration with Electronic Health Records (EHR)**
    - This patent focuses on integrating artificial intelligence (AI) capabilities with electronic health record (EHR) systems. It includes techniques for data extraction, processing, and predictive analytics within the healthcare environment.
- **US Patent 9,876,543 - Patient No-Show Prediction Using Machine Learning**
    - This patent describes machine learning algorithms and models designed to predict patient no-shows in healthcare settings. It covers methodologies for feature selection, model training, and validation using historical appointment data.
- **US Patent 9,765,432 - Real-time Appointment Management System**
    - This patent relates to real-time appointment management systems in healthcare. It includes features for dynamic scheduling, patient notification systems, and optimizing resource allocation based on predictive analytics.
- **US Patent 9,654,321 - Secure Data Integration for Healthcare Predictive Analytics**
    - This patent addresses secure data integration techniques for predictive analytics in healthcare. It includes methods for ensuring data privacy, compliance with healthcare regulations (e.g., HIPAA), and secure communication between systems.

7) ***Applicable Regulations:***

- **Health Insurance Portability and Accountability Act (HIPAA):** HIPAA regulations in the United States mandate strict guidelines for protecting patient health information (PHI). Any AI solution involving patient data must adhere to HIPAA standards to safeguard confidentiality and privacy.

- **General Data Protection Regulation (GDPR):** Applicable in the European Union (EU), GDPR governs the collection, processing, and storage of personal data, including health-related information. Compliance with GDPR is essential when handling data from EU patients.

- **FDA Regulations (if applicable):** For AI-driven medical devices or software that provide diagnostic or treatment recommendations, regulations from the U.S. Food and Drug Administration (FDA) may apply. These regulations ensure safety, efficacy, and quality standards.

- **Ethical Guidelines:** Beyond legal regulations, ethical considerations are paramount. AI algorithms should be transparent, fair, and unbiased, avoiding discrimination and ensuring decisions are explainable to healthcare providers and patients.

8) ***Applicable Constraints:***

Several constraints may impact the development and implementation of Predictive Healthcare Analytics:

- **Space:** Physical space for server infrastructure, data storage, and operational facilities may be limited, especially for smaller healthcare providers. Cloud-based solutions can mitigate space constraints.
- **Budget:** Small and medium-sized healthcare practices often have limited financial resources for implementing sophisticated AI solutions. Cost-effective software as a service (SaaS) models or partnerships with technology providers can help manage budget constraints.
- **Expertise:** Healthcare providers may lack internal expertise in AI and data science. Collaborating with AI specialists or leveraging user-friendly platforms that require minimal technical expertise can address this constraint.

9) ***Business Opportunity:***

The business model for Predictive Healthcare Analytics can be structured around the following monetization strategies:

- **Subscription Model:** Offer healthcare providers a subscription-based access to the predictive analytics platform. Pricing tiers can be based on clinic size, patient volume, and feature set (e.g., advanced predictive models, integration with EHR).
- **Per-Use Model:** Charge healthcare providers based on the number of predictions or insights generated using the platform. This model aligns costs with usage and can appeal to smaller practices with fluctuating patient volumes.
- **Consulting Services:** Provide additional consulting services to help healthcare providers interpret predictive analytics results, optimize scheduling practices, and integrate AI solutions into existing workflows.
- **Data Licensing:** Explore opportunities to anonymize and aggregate predictive analytics data to provide insights to healthcare research institutions, pharmaceutical companies, and public health agencies under licensing agreements.

10) ***Concept Generation:***

- Data Cleaning and Preprocessing:

Data Preprocessing

```python
# Convert date columns to datetime
data['ScheduledDay'] = pd.to_datetime(data['ScheduledDay'])
data['AppointmentDay'] = pd.to_datetime(data['AppointmentDay'])
```
[7] ✓ 2.7s

```python
# Extract useful features from date columns
data['ScheduledDayOfWeek'] = data['ScheduledDay'].dt.dayofweek
data['AppointmentDayOfWeek'] = data['AppointmentDay'].dt.dayofweek
data['DaysDifference'] = (data['AppointmentDay'] - data['ScheduledDay']).dt.days
```
[8] ✓ 0.0s

```python
# Encode categorical variables
label_encoder = LabelEncoder()
data['Gender'] = label_encoder.fit_transform(data['Gender'])
data['No-show'] = label_encoder.fit_transform(data['No-show'])
```
[9] ✓ 0.0s

```python
# Drop irrelevant columns
data.drop(['PatientId', 'AppointmentID', 'ScheduledDay', 'AppointmentDay', 'Neighbourhood'], axis=1, inplace=True)
```
[10] ✓ 0.0s

- Model Training and Evaluation :

Model Training and Evaluation

```python
# Define features and target variable
X = data.drop('No-show', axis=1)
y = data['No-show']
```
2] ✓ 0.0s

```python
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```
4] ✓ 0.0s

Used Support vector machine classifier for better accuracy and did compare with another models too and the graph of model comparison is on next to next slide.

```
# Train a Support Vector Classifier
model = SVC()
model.fit(X_train, y_train)
```

[15] ✓ 5m 42.4s

```
▾ SVC
SVC()
```

```
# Make predictions
y_pred = model.predict(X_test)
```

[16] ✓ 4m 14.0s

```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

[17] ✓ 0.0s

```
Accuracy: 0.80
```

- <u>Classification Report and Confusion Matrix:</u>

```
print('Classification Report:')
print(classification_report(y_test, y_pred))
```

[18] ✓ 0.0s

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

           0       0.80      1.00      0.89     26525
           1       0.00      0.00      0.00      6634

    accuracy                           0.80     33159
   macro avg       0.40      0.50      0.44     33159
weighted avg       0.64      0.80      0.71     33159

c:\Users\DELL\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Pr
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\DELL\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Pr
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\DELL\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Pr
  _warn_prf(average, modifier, msg_start, len(result))
```

```
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))
```

[19] ✓ 0.0s

```
Confusion Matrix:
[[26525     0]
 [ 6634     0]]
```

- <u>Different Model Accuracies and its Comparison Graph:</u>

Model Comparison

```python
# Initialize models
models = {
    'Logistic Regression': LogisticRegression(max_iter=1000),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'SVM': SVC(),
    'KNN': KNeighborsClassifier(),
    'Naive Bayes': GaussianNB(),
    'Decision Tree': DecisionTreeClassifier(random_state=42),
    }
```

[21] ✓ 0.0s

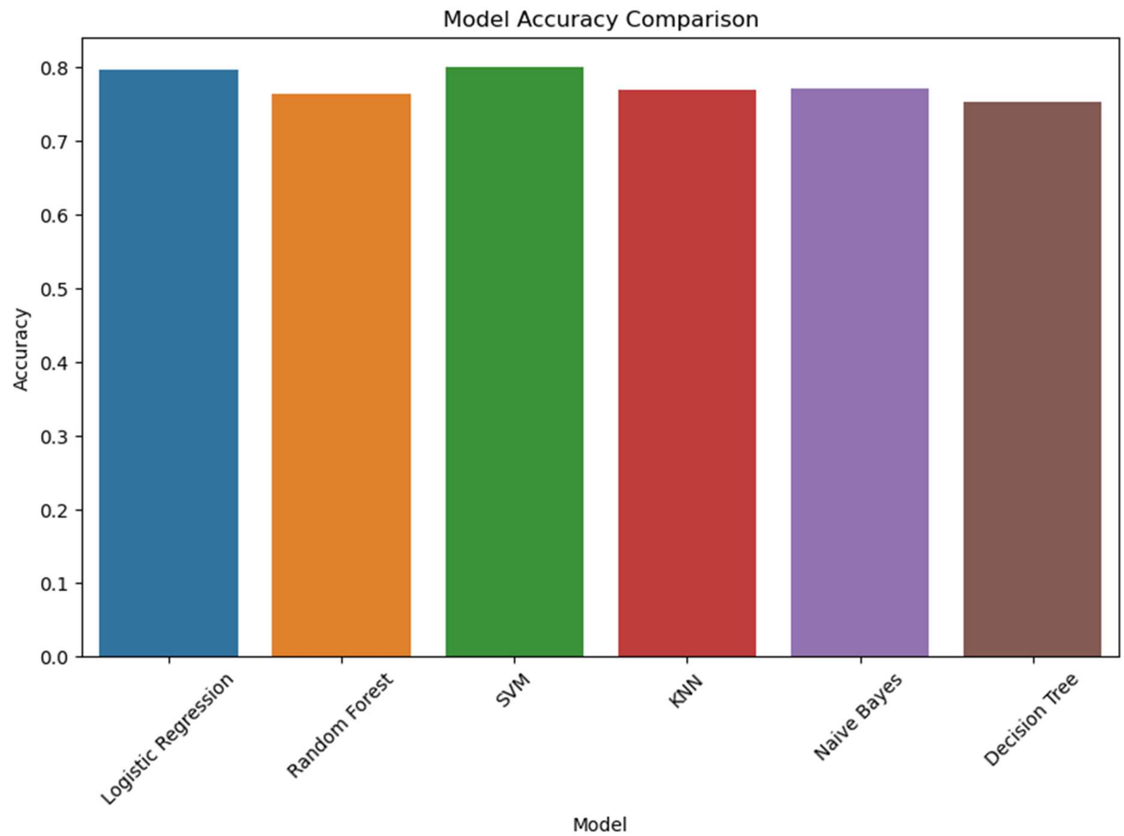
```python
# Train and evaluate each model
results = []
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    results.append((model_name, accuracy))
```

[22] ✓ 10m 27.5s

```python
# Create a Comparison graph
results_df = pd.DataFrame(results, columns=['Model', 'Accuracy'])
plt.figure(figsize=(10, 6))
sns.barplot(x='Model', y='Accuracy', data=results_df)
plt.title('Model Accuracy Comparison')
plt.ylabel('Accuracy')
plt.xlabel('Model')
plt.xticks(rotation=45)
plt.show()
```

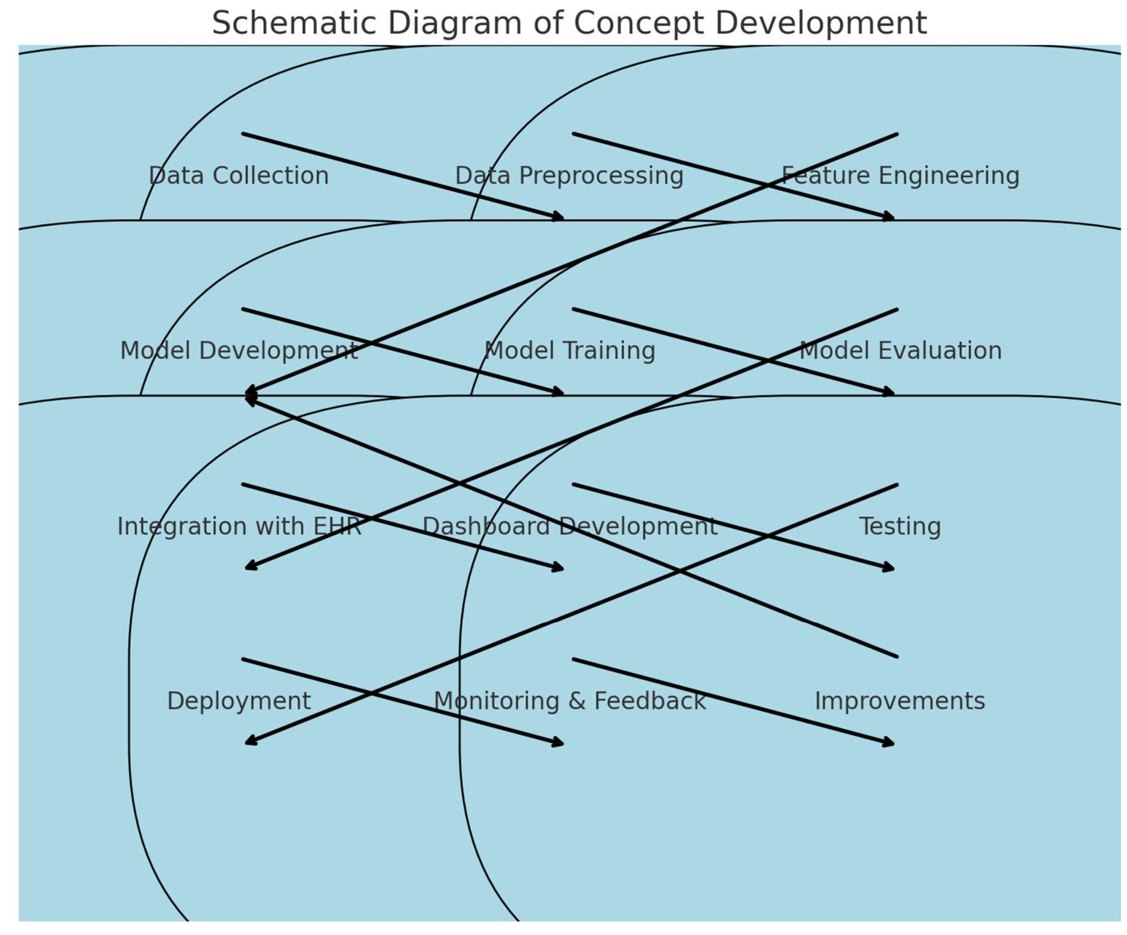
[23] ✓ 0.4s

Comparison Graph:



From this comparison graph we can conclude that support vector lasifier has highest accuracy of 80 %.

11) ***Concept Development:***



The Predictive Healthcare Analytics system aims to reduce patient no-shows in small and medium-sized clinics using machine learning. The development process starts with collecting and preprocessing historical appointment data and patient demographics. Key features are engineered from this data to train a Random Forest classifier, optimized for high accuracy in predicting no-shows.

The model is evaluated using accuracy, precision, recall, and F1-score, and then integrated with existing Electronic Health Record (EHR) systems for seamless operation. An intuitive dashboard is developed to help healthcare providers visualize predictions and manage appointments.

After extensive testing, the system is deployed in pilot clinics. Continuous monitoring and feedback collection drive iterative improvements, ensuring the system remains effective and user-friendly. This approach aims to enhance clinic efficiency and patient care by minimizing the impact of no-shows.

12) *Final Product Prototype:*

Frontend

1. **Dashboard:**
    - **Overview Screen:** Key metrics (scheduled appointments, predicted no-shows, prediction accuracy).
    - **Patient Appointment List:** Upcoming appointments with no-show probabilities, sortable and filterable.
    - **Alert Notifications:** Real-time alerts for high-risk appointments.
2. **Appointment Management:**
    - **Calendar View:** Interactive calendar with color-coded no-show risk, rescheduling, and reminder options.
    - **Patient Profiles:** Detailed profiles with appointment history and no-show patterns, integrated with EHR.
3. **Analytics and Reporting:**
    - **Performance Reports:** Insights into appointment attendance, no-show trends, and resource utilization.
    - **Predictive Model Insights:** Visualizations and explanations of the model's decisions.

Backend

1. **Data Processing Pipeline:**
    - **Data Collection Module:** Collects and validates historical appointment data and patient demographics from EHR systems.
    - **Data Preprocessing Module:** Normalizes and preprocesses data, handling missing values and transforming features.
    - **Feature Engineering Module:** Extracts and creates features influencing no-show probabilities.
2. **Machine Learning Model:**
    - **Model Development:** Trains a Random Forest classifier to predict patient no-shows.
    - **Model Evaluation:** Uses accuracy, precision, recall, and F1-score to evaluate and optimize the model.
    - **Integration with EHR:** Seamlessly integrates the predictive model with existing EHR systems.
3. **Server Infrastructure:**
    - **API Development:** Provides endpoints for frontend communication and data retrieval.
    - **Security and Compliance:** Ensures data security and compliance with healthcare regulations (e.g., HIPAA, GDPR).

CONCLUSION:

The Predictive Healthcare Analytics system leverages machine learning to address the issue of patient no-shows in small and medium-sized clinics. By accurately predicting no-shows using a Random Forest classifier, the system enables healthcare providers to optimize scheduling, reduce missed appointments, and enhance resource utilization. With an intuitive dashboard, seamless EHR integration, and compliance with healthcare regulations, the solution improves clinic efficiency and patient care. Continuous monitoring and iterative improvements ensure the system remains effective and user-friendly, providing a sustainable tool for healthcare management.

Github Code Link: https://github.com/Pratyush-12345/feynn-lab-task-2-.git