



MACHINE LEARNING

IN HEALTHCARE

PROJECT BY:

AKSHITH SARAVANAN, SUMEDHA KOMAWAR,
ADITYA KUMAR, SAURAV MUKHERJEE,
MANVITH KOTHAPALLI




MEDICINE ROUTE



PREDICTING THE MEAN
COST OF A DISCHARGE



WHY WE CHOSE THIS TRACK



As students passionate about both technology and real-world impact, we wanted to solve a problem that truly matters. We realized that by predicting hospital discharge costs, we could help hospitals, doctors, and patients plan better and ease future burdens. Our machine learning model is our way of giving back using AI to support healthcare systems and make a real difference in people's lives. We believe that even as students, we can build tools today that create a stronger, smarter future tomorrow.



LINKS



<https://github.com/akshithsaravanan/DubsTech-Health-ML-2025>



<https://colab.research.google.com/drive/IG-M-d6rRM84h3ZrcDMDJMFvIrJFWVfPh?usp=sharing>

TABLE OF CONTENT

01

**ML
MODEL**

02

**PRELIMINARY
RESEARCH**

03

**OUR
RESULTS**

04

**OUR
CONCLUSION**



ABSTRACT

As we explored the dataset, we were surprised by its large size and complexity. Initially, we experimented with different machine learning models such as Linear Regression and Polynomial Regression (degree = 2), but these models resulted in low prediction accuracy. After further experimentation, we implemented a CatBoost Regressor, which significantly improved our results, achieving a high accuracy and a very low margin of error. This demonstrated the importance of selecting the right model architecture, especially when working with large, mixed-type datasets like ours.



01

ML MODEL

PREPARING THE DATA AND CLEANING THE DATA

1) First mounted the dataset from Google Drive

```
[ ] from google.colab import drive  
    drive.mount('/content/drive')
```

2) Verified the dataset to ensure that it had been loaded correctly before proceeding further.

```
[ ] # The full path  
    df = pd.read_csv('/content/drive/MyDrive/Hospital_Inpatient_Discharges__SPARCS_De-Identified__Cost_Transparency__Beginning_2009_20250426 (1).csv')  
  
    # Viewing shape  
    print(df.shape)
```

```
(1192827, 14)
```


PREPARING THE DATA AND CLEANING THE DATA

3) We explored the dataset to check the column names, data types, and non-null counts for each column. Additionally, we obtained basic statistical summaries (mean, median, min, max, standard deviation) for all numeric columns and calculated the number of missing (null) values in each column, allowing us to quickly detect incomplete or problematic data that may need cleaning.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1192827 entries, 0 to 1192826
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Year                                1192827 non-null  int64
1   Facility Id                        1192827 non-null  int64
2   Facility Name                      1192827 non-null  object
3   APR DRG Code                      1192827 non-null  int64
4   APR Severity of Illness Code      1192827 non-null  int64
5   APR DRG Description               1192827 non-null  object
6   APR Severity of Illness Description 1192617 non-null  object
7   APR Medical Surgical Code         1192348 non-null  object
8   APR Medical Surgical Description   1192827 non-null  object
9   Discharges                       1192827 non-null  object
10  Mean Charge                      1192827 non-null  object
11  Median Charge                    1192827 non-null  object
12  Mean Cost                       1192827 non-null  object
13  Median Cost                      1192827 non-null  object
dtypes: int64(4), object(10)
memory usage: 127.4+ MB
```

```
[ ] # Check column names and data types
df.info()

# See basic statistics
df.describe()

# See missing values
df.isnull().sum()
```

Year	0
Facility Id	0
Facility Name	0
APR DRG Code	0
APR Severity of Illness Code	0
APR DRG Description	0
APR Severity of Illness Description	210
APR Medical Surgical Code	479
APR Medical Surgical Description	0
Discharges	0
Mean Charge	0
Median Charge	0
Mean Cost	0
Median Cost	0

dtype: int64

PREPARING THE DATA AND CLEANING THE DATA

4) Here we cleaned the column names.

```
[ ] # Cleaning column names
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
df.head(2)
```

5) We converted the cost and charge columns to the correct numerical format to allow proper mathematical operations and modeling later in the project.

```
[ ] # Converting cost and charge columns to numeric
cols_to_numeric = ['discharges', 'mean_charge', 'median_charge', 'mean_cost', 'median_cost']

# Forcing columns to string first, then remove commas/dollar signs, then converting to float
for col in cols_to_numeric:
    df[col] = df[col].astype(str).str.replace(',', '').str.replace('$', '').astype(float)
```

PREPARING THE DATA AND CLEANING THE DATA

6) We checked again for missing values after cleaning the data. This is to confirm that our data is fully prepared for the next steps, like model training.

```
[8] # Checking again for missing values
df.isnull().sum()
```

```
year      0
facility_id  0
facility_name  0
apr_drg_code  0
apr_severity_of_illness_code  0
apr_drg_description  0
apr_severity_of_illness_description  210
apr_medical_surgical_code  479
apr_medical_surgical_description  0
discharges  0
mean_charge  0
median_charge  0
mean_cost  0
median_cost  0
dtype: int64
```

PREPARING THE DATA AND CLEANING THE DATA

```
[ ] df = df.dropna()

# Checking again
print(df.shape)
df.isnull().sum()
```

→ (1192348, 14)

	0
year	0
facility_id	0
facility_name	0
apr_drg_code	0
apr_severity_of_illness_code	0
apr_drg_description	0
apr_severity_of_illness_description	0
apr_medical_surgical_code	0
apr_medical_surgical_description	0
discharges	0
mean_charge	0
median_charge	0
mean_cost	0
median_cost	0

dtype: int64

7) We eliminated any rows that contained missing (null) values to ensure that our dataset was completely clean. Moreover, we confirmed that there were zero missing values across all columns, guaranteeing that the dataset is fully clean and ready for reliable model training.

PREPARING THE DATA AND CLEANING THE DATA

```
[ ] # Checking shape
print("Shape of dataset:", df.shape)

# printing the number of rows
print("Number of rows:", df.shape[0])

# And number of columns
print("Number of columns:", df.shape[1])
```

```
⇒ Shape of dataset: (1192348, 14)
   Number of rows: 1192348
   Number of columns: 14
```

8) We confirmed the dimensions of our dataset after cleaning, so that we were aware of the final dataset size before proceeding to data exploration and modelling.

```
[ ] # Checking for negative costs or charges
print((df['mean_cost'] < 0).sum())
print((df['median_cost'] < 0).sum())
print((df['mean_charge'] < 0).sum())
print((df['median_charge'] < 0).sum())
print((df['discharges'] < 0).sum())
```

```
⇒ 0
   0
   0
   0
   0
   0
```

9) We validated the financial and discharge-related columns to ensure that there were no negative values, which would be illogical in the context of healthcare billing.

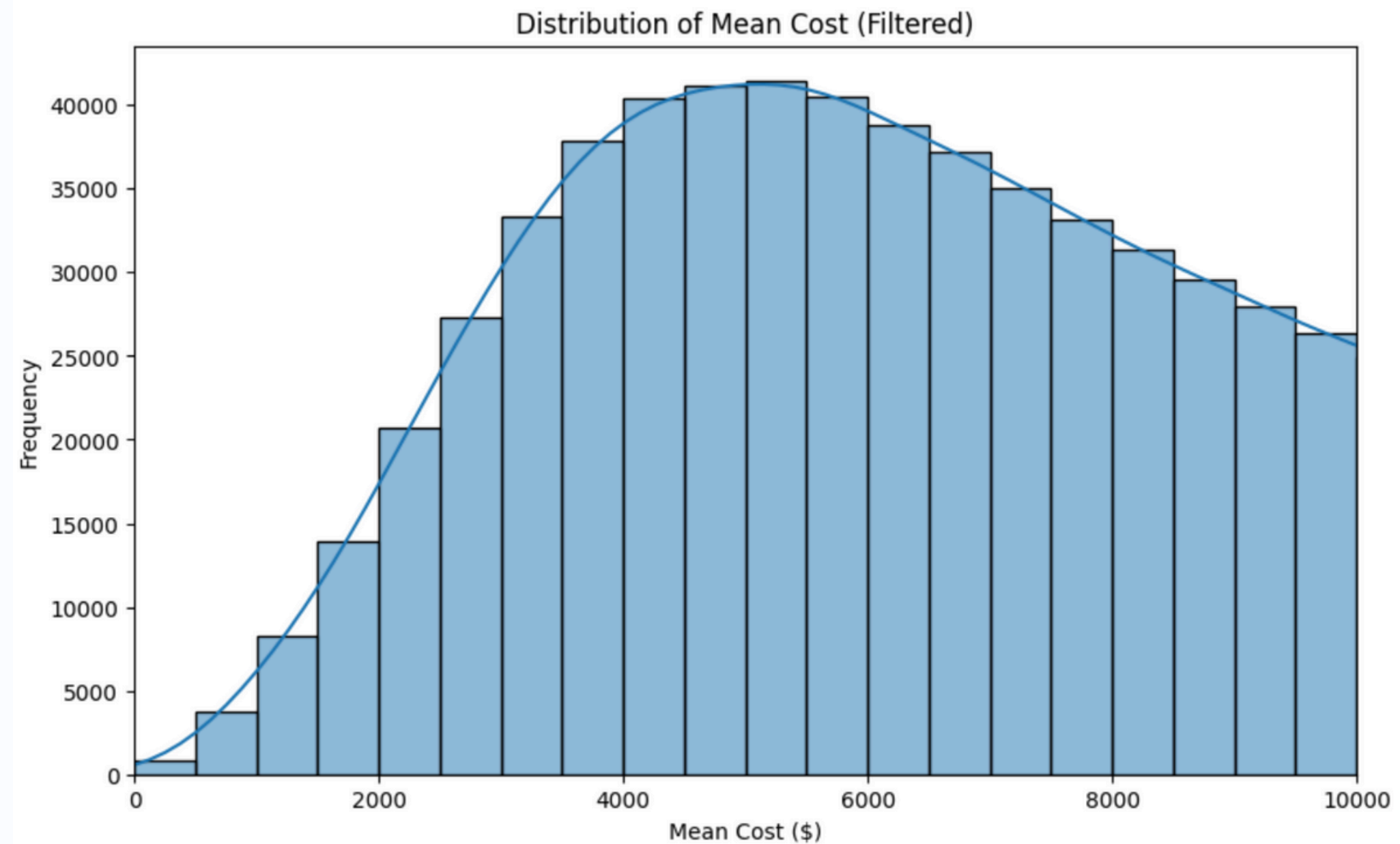


02

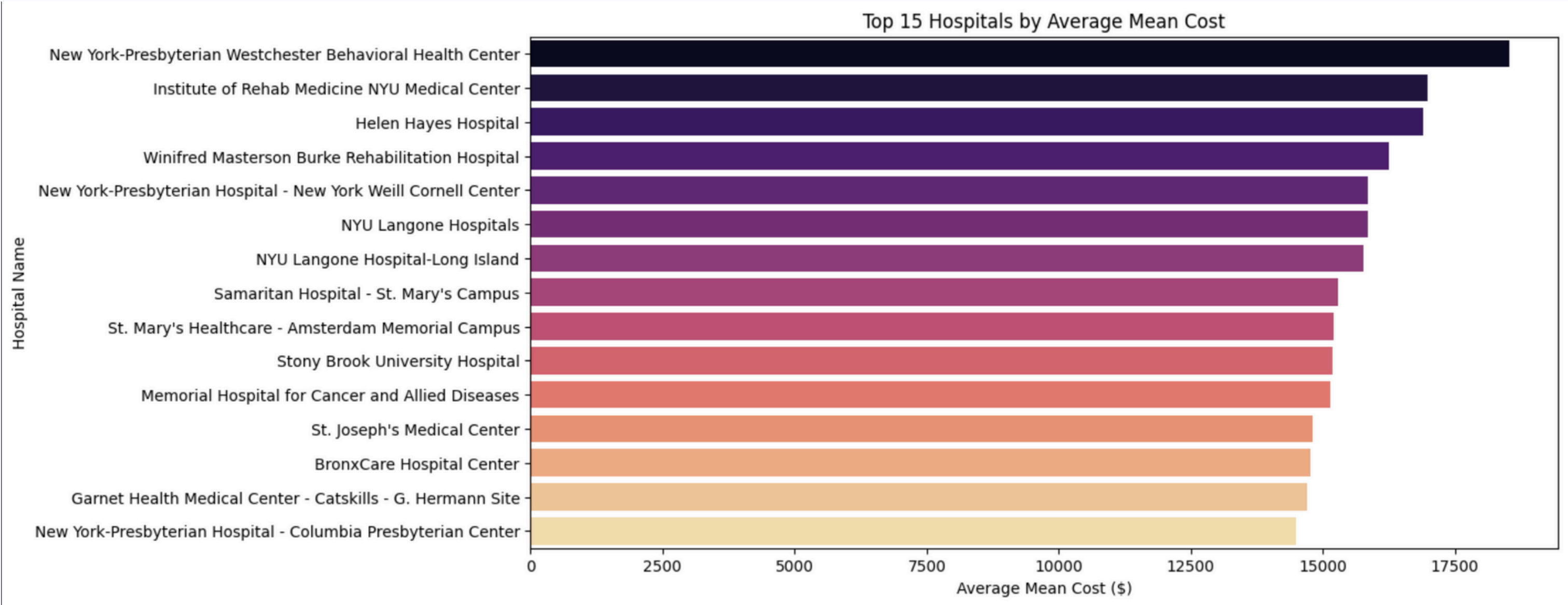
PRELIMINARY RESEARCH

STEP 1 - GENERATING BASIC STATISTICAL SUMMARIES FOR KEY COLUMNS

Before we started creating our machine learning model, we formatted our existing data into different graphs to understand the patterns and better understand it.

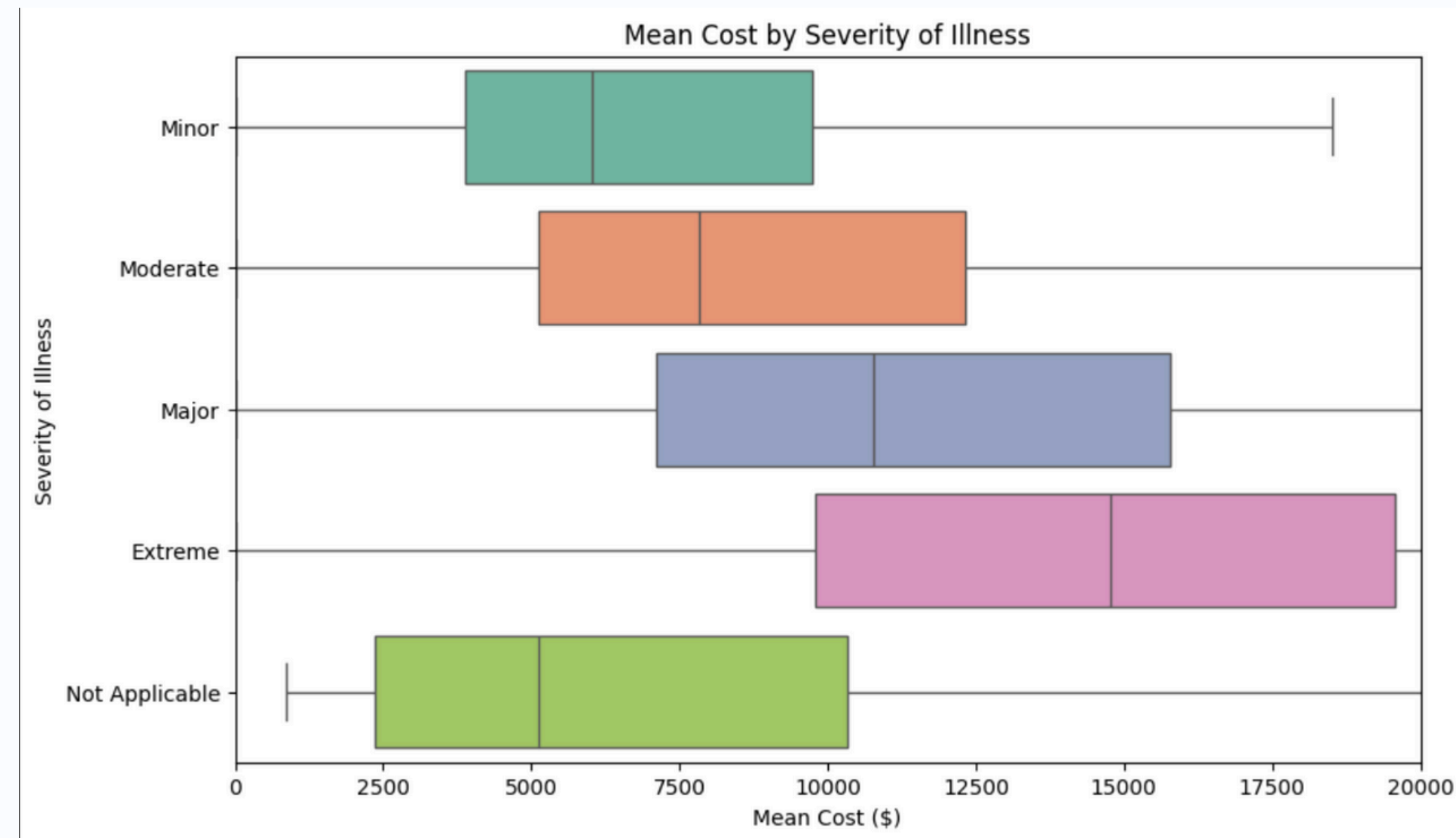


STEP 1 - GENERATING BASIC STATISTICAL SUMMARIES FOR KEY COLUMNS



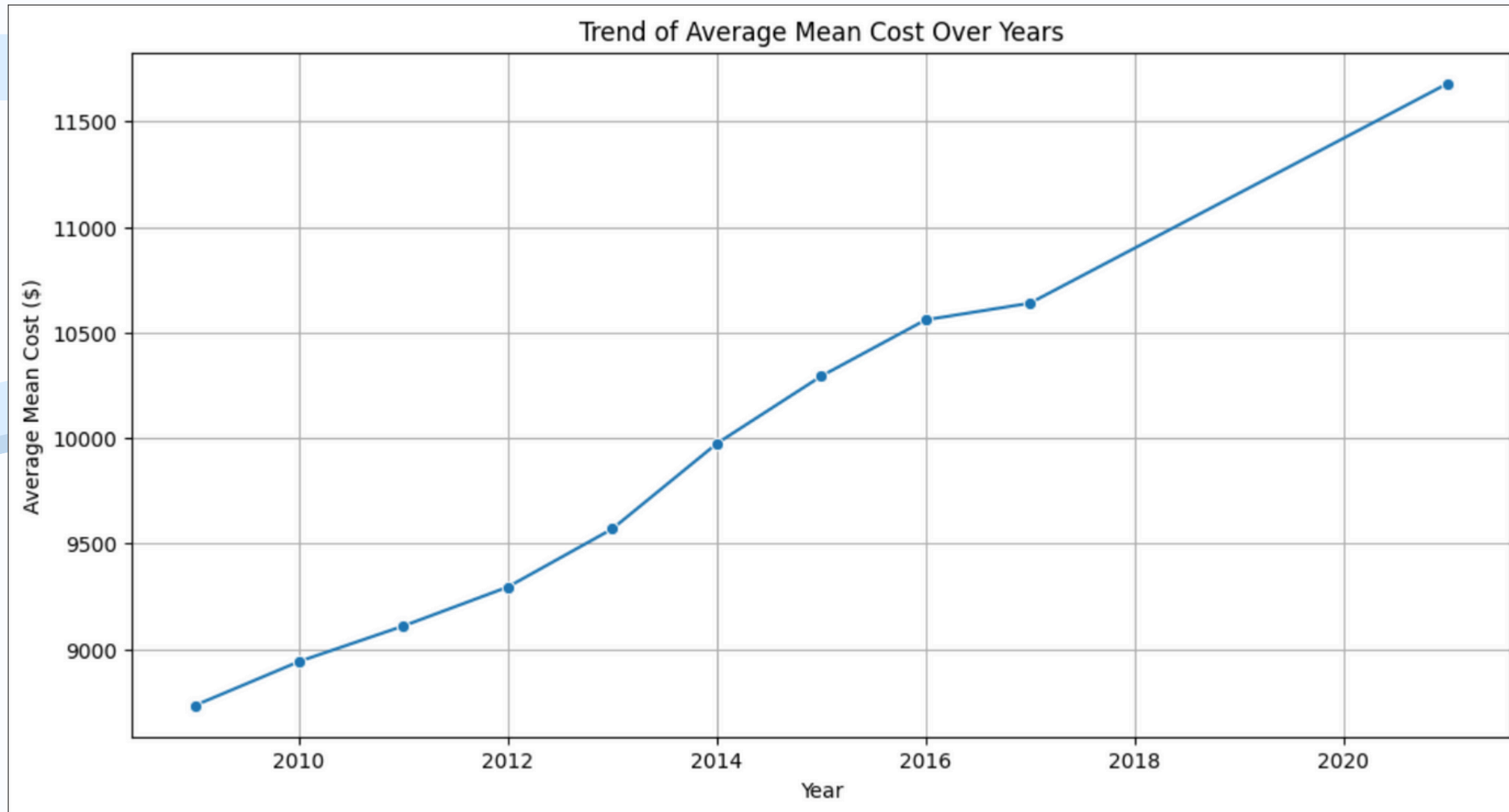
We looked at the top 15 hospitals by average mean cost to understand around what value our prediction should be.

STEP 1 - PREPARING THE DATA AND CLEANING THE DATA



Next, we looked at some box and whisker plots to understand how the mean cost was spread according to the severity of illness. The box representing extreme severity showed a wide spread, indicating that the data was highly varied and suggesting the presence of several outliers. We inferred that, since extreme cases are often experimental, they tend to produce more outliers.

STEP 1 - PREPARING THE DATA AND CLEANING THE DATA



We created a trend analysis curve and saw that there was a steep increase in the data from 2012 to 2017. This helped us come up with our conclusion that the price of healthcare increased.

STEP 2 - GENERATING BASIC STATISTICAL SUMMARIES USING TABLEAU

<https://public.tableau.com/app/profile/sumedha.komawar/viz/DataHackathondashboard1/Dashboard1?publish=yes>

<https://public.tableau.com/app/profile/sumedha.komawar/viz/DataHackathondashboard2/Dashboard2?publish=yes>

<https://public.tableau.com/app/profile/sumedha.komawar/viz/DataHackathondashboard2/Dashboard3?publish=yes>



OUR RESULTS

```
# FINAL MODEL: High Accuracy CatBoost Model for Mean Cost Prediction
```

```
from catboost import CatBoostRegressor
from sklearn.model_selection import train_test_split
from math import sqrt
```

```
# 1. Select strong features
```

```
selected_features = [
    'facility_name',
    'apr_drg_code',
    'apr_severity_of_illness_description',
    'apr_medical_surgical_description',
    'year',
    'mean_charge',
    'median_charge'
]
```

```
target_variable = 'mean_cost' # We are predicting mean cost
```

```
# 2. Prepare the data
```

```
model_df = df_filtered[selected_features + [target_variable]].dropna()
X = model_df[selected_features]
y = model_df[target_variable]
```

```
# 3. Identify categorical features
```

```
cat_features = [0, 1, 2, 3, 4]
```

```
# 4. Train-test split
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
# 5. Build and train the model
```

```
model = CatBoostRegressor(
    iterations=1200,
    depth=10,
    learning_rate=0.03,
    loss_function='RMSE',
    random_seed=42,
    cat_features=cat_features,
    verbose=100
)
```

```
model.fit(X_train, y_train)
```

```
# 6. Evaluate the model
```

```
train_score = model.score(X_train, y_train)
test_score = model.score(X_test, y_test)
```

```
print(f"\n Final Train Score: {train_score:.4f}")
print(f" Final Test Score: {test_score:.4f}")
```

```
# 7. Margin of Error Calculation
```

```
n = len(y_test)
z = 1.96 # 95% confidence interval (standard z-score)
p = test_score
```

```
margin_of_error = z * sqrt(p * (1 - p) / n)
```

```
print(f" Margin of Error: ±{margin_of_error:.4f}")
```

```
# 8. Predict for 2025
```

```
X_2025 = X.copy()
X_2025['year'] = 2025
```

```
predicted_mean_costs_2025 = model.predict(X_2025)
average_predicted_mean_cost_2025 = predicted_mean_costs_2025.mean()
```

```
print(f"\n🎯 Predicted Average Mean Cost for 2025 (Used Full Dataset): ${average_predicted_mean_cost_2025:.2f}")
```

We developed our machine learning model, a high-accuracy CatBoost Regressor, to reliably forecast hospital discharge costs using key features and successfully predicted the average mean cost for 2025.

```
Final Train Score: 0.9377  
Final Test Score: 0.9341  
Margin of Error:  $\pm 0.0011$ 
```

```
🎯 Predicted Average Mean Cost for 2025 (Used Full Dataset): $11106.70
```

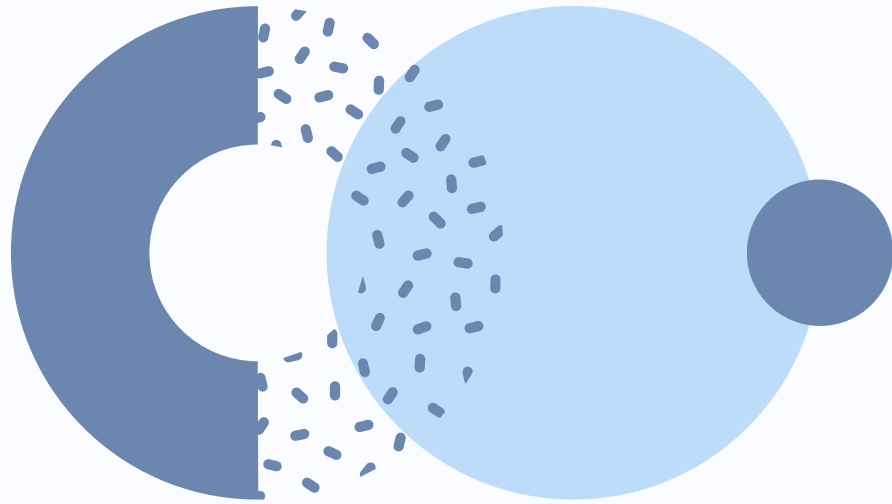
Our final machine learning model demonstrated strong predictive performance, achieving a high training score of 0.9377, a high testing score of 0.9341, and a low margin of error of ± 0.0011 at a 95% confidence interval. These metrics show that our model generalizes well to unseen data with minimal overfitting. The model took approximately 35 minutes to run, which is reasonable given the large size of the dataset, consisting of over one million records. Extensive data cleaning was performed, including the removal of major outliers to ensure the reliability of the predictions. After model training and evaluation, we predicted that the average mean cost for a hospital discharge in 2025 would be approximately \$11,106.70. With the strong performance and low margin of error, we are highly confident in the model's ability to support accurate healthcare budget planning and resource allocation.

Finding Area	Summary insight
Discharges over time	Discharges consistently decreased, sharpest drop 2012–2017. Likely due to healthcare costs and economic pressures.
Facility vs Discharges	University Hospitals & Westchester Medical Center have the highest discharges. Smaller hospitals lost patients.
Charges vs Cost Analysis	Charges are significantly higher than costs across facilities; possibly discouraging uninsured patients.
Hospital Gap Margins	Westchester Medical Center and University Hospital show the biggest charge-cost gaps.
Discharges by Severity	Minor and moderate severity cases dominate hospital discharges. Extreme cases remain rare.
Surgical vs Medical Discharges	Medical discharges dominate but decline faster; elective surgeries stay steady.
Low- severity medical trends	Sharp decline from 1.5M to 1M discharges for low-severity medical cases, linked to rising costs.

- Problems (costs, patient avoidance, hospital financial challenges)
- Improvement (minor illnesses dominate = better health outcomes)
- Mixed/neutral trend (surgical discharges steady, medical declines sharply)



OUR CONCLUSION



KEY TAKEAWAYS

High Model Performance and Strong Predictions

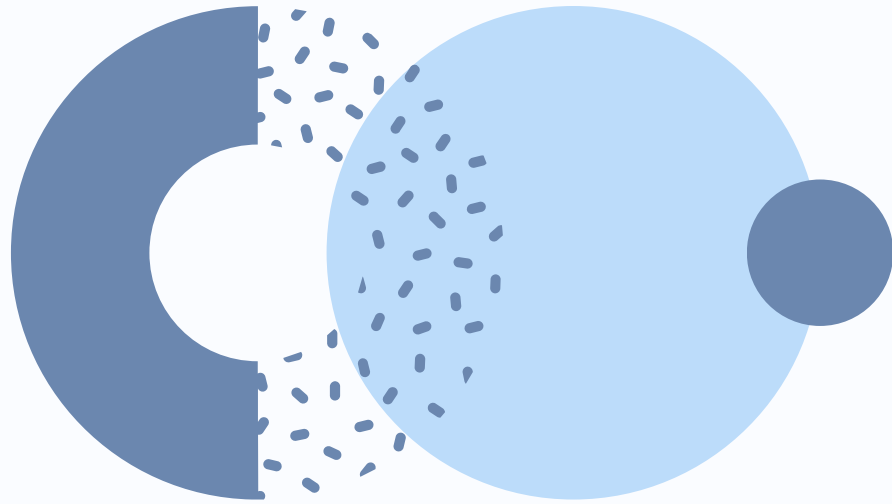
- Achieved a train score of 0.9377 and test score of 0.9341



- Very low margin of error: ± 0.0011 (95% confidence interval)



- Predicted 2025 average hospital discharge cost \approx \$11,106.70



KEY TAKEAWAYS

Robust Data Handling and Healthcare Impact

- Cleaned over 1 million records and Addressed outliers to boost model reliability



- Supports budget forecasting and resource planning



- Empowers data-driven healthcare decisions

The background is a light blue gradient. It features several medical-themed illustrations: a stethoscope on the right side, a pill bottle with a white cross on its label at the top center, a blister pack of pills on the top left, a hand holding a clipboard on the bottom left, and a rounded square icon with a cross on the bottom left. There are also four plus signs scattered across the background.

THANK YOU