Cellula Technologies

Hotel Booking Status Prediction Using Machine Learning & Flask

Presentation: 3rd Task

Group : ML_Group2

Submitted By: Dalia Nasser

1. Modeling & Evaluation

Multiple classification models were evaluated, with **Random Forest achieving the highest test** set accuracy at **86.9%**, closely followed by **XGBoost at 86.6%**, while **Logistic Regression trailed at 80.2%**, demonstrating the advantage of ensemble methods for this prediction task.

Which classification model should be used for booking status prediction?

Random Forest

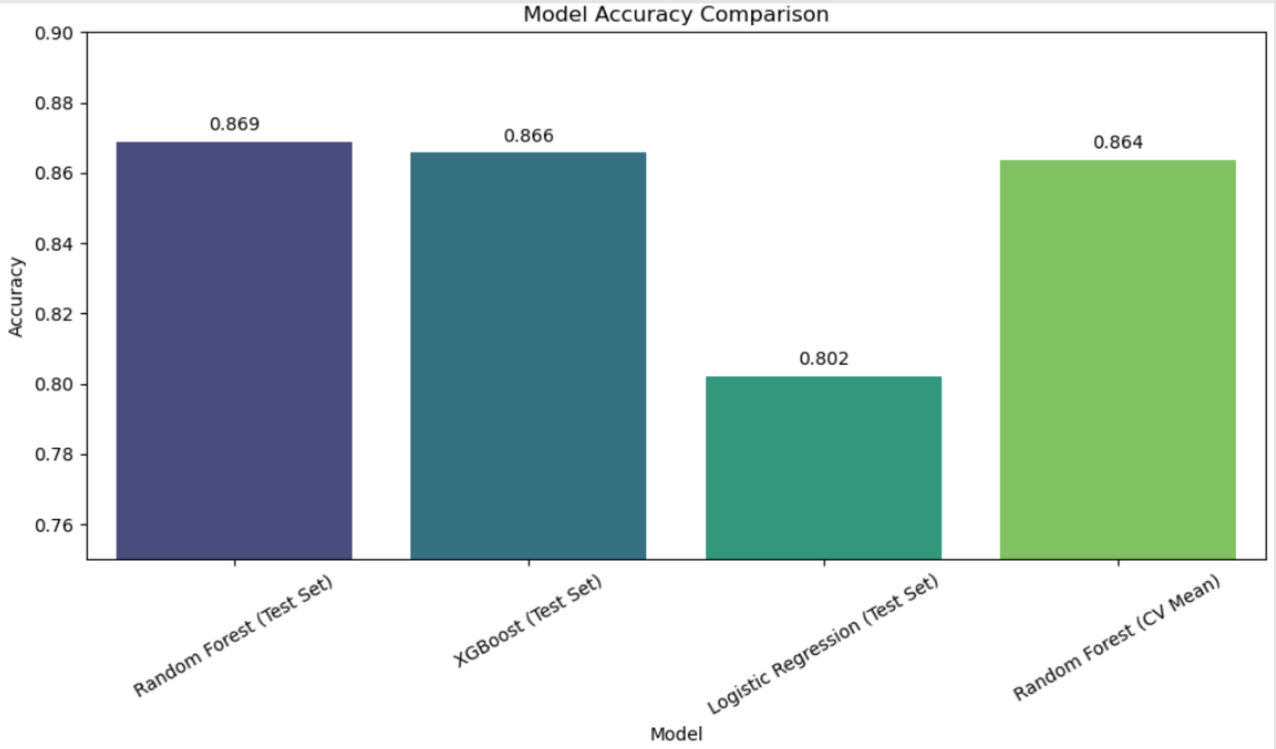
Offers high accuracy and robustness but can be computationally intensive.

Logistic Regression

Simple and interpretable but may not capture complex relationships.

XGBoost

High performance and scalability but requires careful tuning.



Confusion Matrix & Classification Report

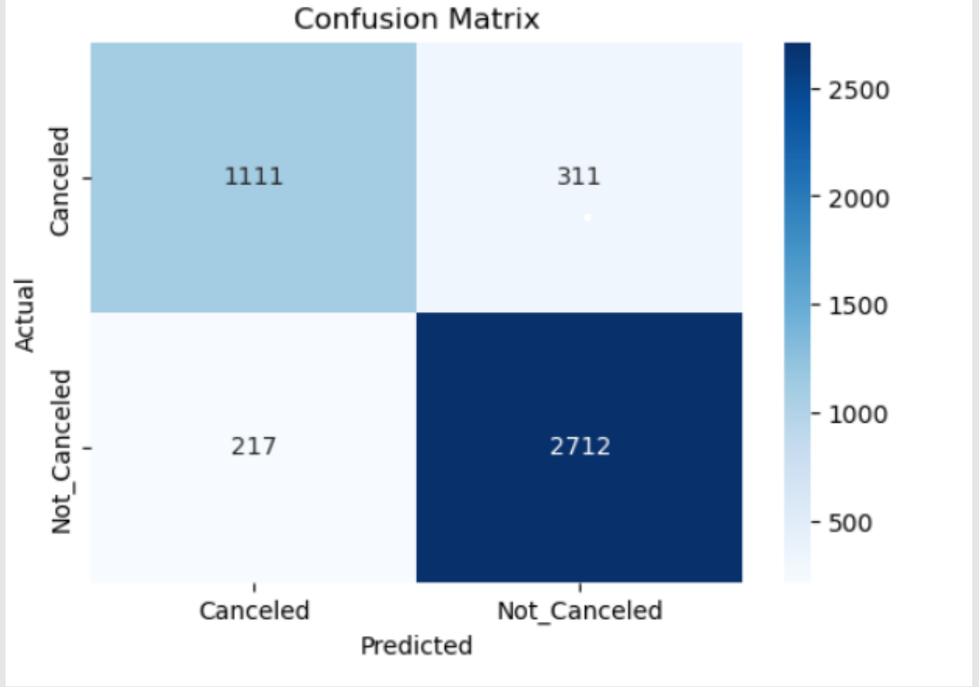
The model accurately predicted **1,111 canceled bookings** and **2,712 not-canceled bookings**, with relatively low **false positives (311)** and **false negatives (217).** This indicates the model performs well in distinguishing between the two classes with minimal misclassifications.

➢ Classification Report Insight:

- The model achieved a high **overall accuracy of 88%,** showing strong performance.
- It has better performance on the **Not_Canceled class** (Precision: 0.90, Recall: 0.93) compared to the Canceled class.
- The **F1-score** for both classes (0.81 for Canceled, 0.91 for Not_Canceled) indicates a balanced trade-off between precision and recall.
- The **macro average F1-score is 0.86**, meaning the model maintains solid general performance across both classes.

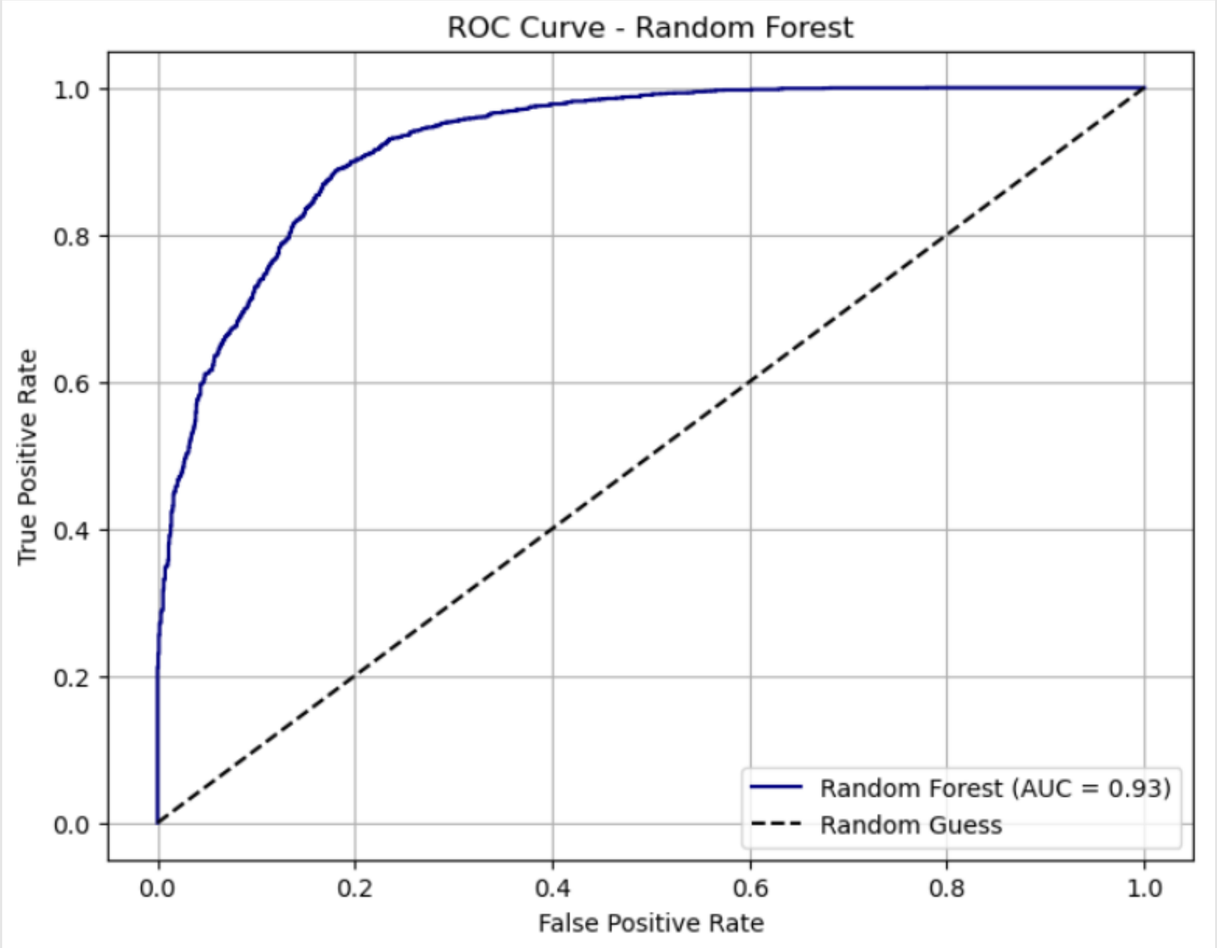
Classification Report:

	precision	recall	f1-score	support
Canceled	0.84	0.78	0.81	1422
Not_Canceled	0.90	0.93	0.91	2929
accuracy			0.88	4351
macro avg	0.87	0.85	0.86	4351
weighted avg	0.88	0.88	0.88	4351



2. ROC Curve & AUC Evaluation

The ROC Curve demonstrates the model's ability to distinguish between the two classes (Canceled vs Not_Canceled). The **Random Forest model** achieved a high **AUC score of approximately 0.93**, indicating excellent performance. The curve is well above the diagonal "random guess" line, confirming that the model is effective in correctly identifying positive cases while minimizing false positives.

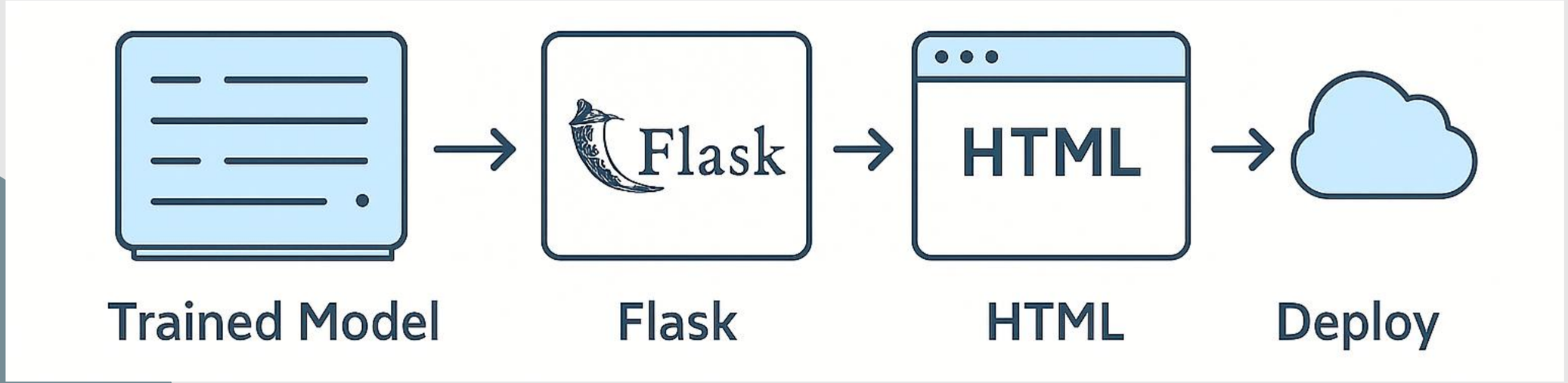


3. Model Saving for Deployment

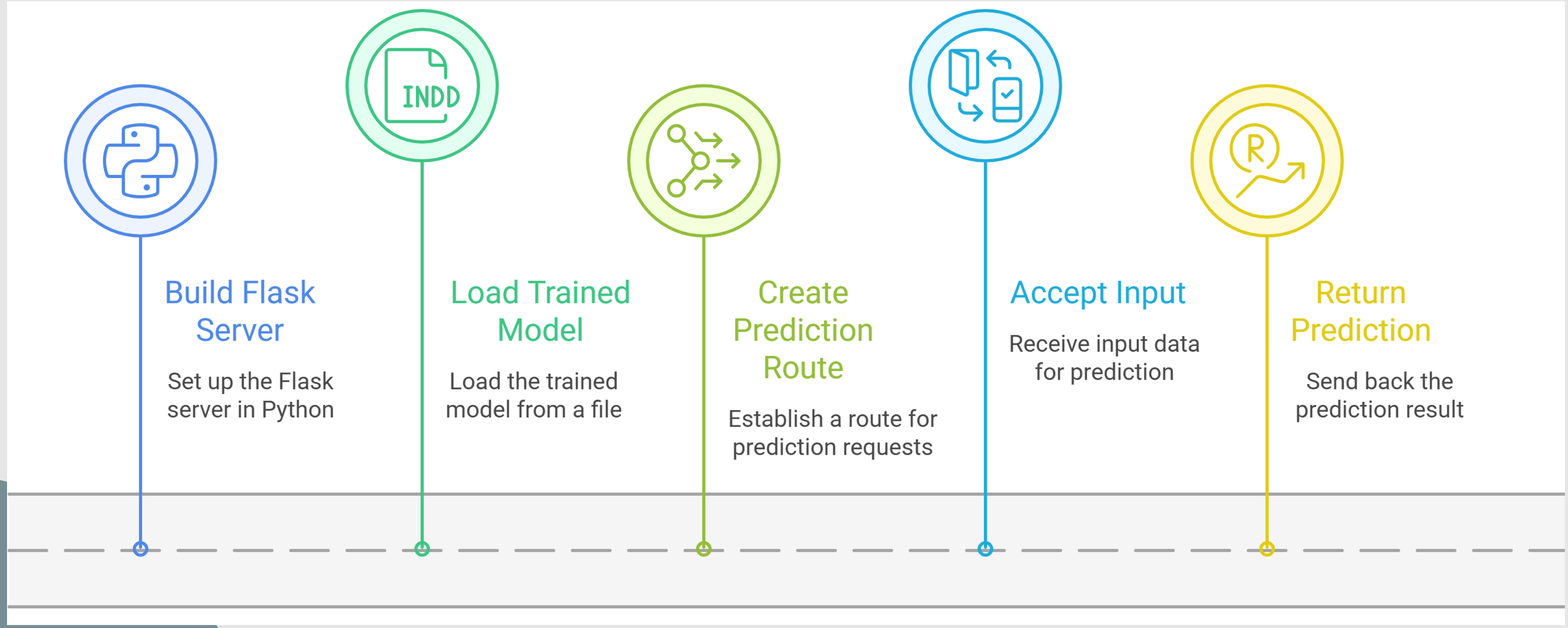
- ✓ The trained model was saved using **joblib** into a file named **rf_model.pkl**.This step is crucial for deploying the model in a flask application, allowing predict be made on new user inputs without retraining the model every time.

- ✓ **Deploying:** means **making it accessible** to others (or to other software) via an **API**.
- ✓ **Flask:** is a lightweight Python web framework for building **APIs**.You wrap your model in a **Flask app**, which:

1. Accepts new data as input (e.g., from a web request)
2. Uses your trained model to **make a prediction**
3. Returns the prediction result in the response



4. Flask Deployment Process



5. Make a Good Interface

1

Build HTML Page

Create a user interface with a form

2

User Enters Details

Input booking information into the form

3

Submit Form

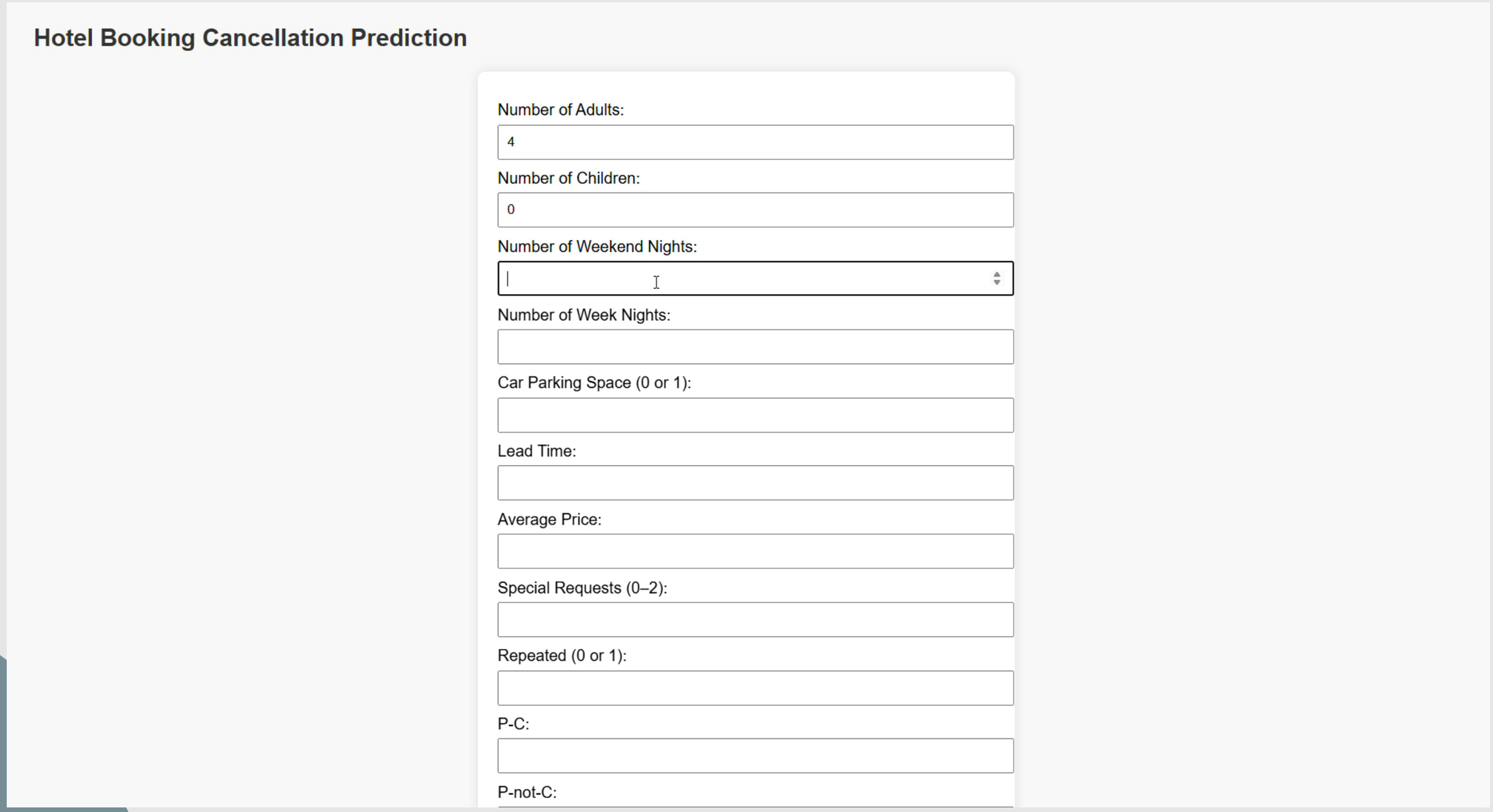
Send the form data for processing

4

See Predicted Result

Display the booking prediction to the user

6. Final Output



Thank You