Deployment Plan - Wikipedia Page Views Forecasting Project

Deployment Method

I would deploy my model using a simple Flask application where a user can select a Wiki page name and then the forecast would be returned for the next day. Since the model predicts future values, I would log each prediction along with the date and page name. When the actual traffic data becomes available, I would compare it to the prediction and calculate MAE. This would allow me to track model accuracy over time and detect when performance begins to decline.

After Deployment - Monitoring

If the model's MAE increases over time or I notice unusual predictions in the logs, I would retrain the model using updated Wikipedia traffic data. I would do this manually at first, but later I could automate it using a scheduled script. The new model would then be re-deployed. I would also check logs weekly to make sure predictions still make sense and nothing is broken.

Other Available Options

There are many ways to deploy ML models. I chose Flask because it is easy to use and perfect for beginners. If the project needed to scale later, I could move it to AWS SageMaker. For this small project, Flask is the most practical choice.

Trade-offs

Choosing Flask is fast and low-cost, but it doesn't have built-in monitoring or automatic retraining. That's okay for now, since I'm manually reviewing logs and calculating MAE. If the app needed to scale, I would move to a cloud platform and set up automatic logging and alerts. I also chose to train one model per page for better accuracy, even though it takes longer. This makes the system more accurate, but slower to scale.

Final Pipeline

This deployment comes after the model is trained and evaluated. The full ML pipeline includes cleaning the data, transforming it into supervised format, training and validating the model, and then using the trained model to make predictions. The Flask app would load the trained model and accept input data to return a forecast. It would also log predictions so I can monitor performance and retrain when needed.