Product Requirements and Specification Document

Project Name

ChurnGuard - Customer Churn Prediction API

Description

ChurnGuard is a FastAPI-based backend service for predicting customer churn using an XGBoost model. It provides RESTful endpoints for predictions, model explanations (via SHAP), and includes interactive API documentation (Swagger UI). The service is containerized with Docker and deployable on Hugging Face Spaces.

---

1. Goals & Objectives

Goal	Description
Predict Customer Churn	Provide accurate churn predictions via API
Explain Model Predictions	Offer SHAP-based explanations for transparency
Easy Integration & Testing	Expose endpoints with Swagger UI for developer access
Portable & Deployable	Support Docker and Hugging Face Spaces deployment

---

2. Core Features

Feature	Description
Prediction Endpoint	Accepts customer data, returns churn probability
Explanation Endpoint	Returns SHAP values for input data
Swagger UI	Interactive API documentation and testing
Docker Support	Dockerfile for containerized deployment
Hugging Face Spaces	Integration for public demo and deployment

---

3. Functional Requirements

3.1 API Endpoints

Endpoint	Method	Request Body	Response	Auth
`/predict`	POST	Customer features	Churn probability (JSON)	None
`/explain`	POST	Customer features	SHAP values (JSON)	None
`/docs`	GET	-	Swagger UI	None

Example Request/Response

```
// POST /predict
{
  "age": 35,
  "tenure": 5,
  "balance": 10000,
  "products": 2,
  "credit_score": 700
}
```

```
// Response
{
  "churn_probability": 0.23
}
```

3.2 Model Integration

- Use pre-trained XGBoost model (binary classification)
- Load model at startup
- Use SHAP for explanations

3.3 Input Validation

- Validate all input fields (type, range)
- Return 400 error for invalid input

---

4. Non-Functional Requirements

Requirement	Specification
Performance	< 500ms response time for typical requests
Scalability	Stateless, supports horizontal scaling
Security	No sensitive data stored; input validation
Documentation	Auto-generated via FastAPI/Swagger
Portability	Dockerized; deployable on Hugging Face Spaces

---

5. Technical Specifications

Component	Technology/Tool	Notes
API Framework	FastAPI	Async, OpenAPI support
ML Model	XGBoost	Pre-trained, binary classification
Model Explainability	SHAP	TreeExplainer for XGBoost

Containerization	Docker	Dockerfile provided
Deployment	Hugging Face Spaces	Public demo, easy sharing
Language	Python 3.8+	

---

6. API Specification

Field	Type	Required	Description
age	int	Yes	Customer age
tenure	int	Yes	Years as customer
balance	float	Yes	Account balance
products	int	Yes	Number of products
credit_score	int	Yes	Credit score
…	…	…	(Extend as per model)

---

7. Deployment & Integration

7.1 Docker

- Provide `Dockerfile` for building and running the API
- Expose port 8000

7.2 Hugging Face Spaces

- Ensure compatibility with Spaces requirements
- Include `app.py` and `requirements.txt`

---

8. Acceptance Criteria

Criteria	Method of Verification
API returns valid churn predictions	Unit/integration tests
SHAP explanations returned for inputs	Unit/integration tests
Swagger UI accessible at `/docs`	Manual test
Docker image builds and runs successfully	Build/test pipeline
Deploys and runs on Hugging Face Spaces	Manual/automated deployment

---

9. Out of Scope

- Frontend UI (beyond Swagger UI)
- User authentication/authorization
- Model training pipeline (assume pre-trained model)

10. Milestones

Milestone	Target Date
API endpoints implemented	Week 1
Model integration & SHAP	Week 2
Dockerization	Week 2
Hugging Face Spaces deployment	Week 3
Documentation & Testing	Week 3

11. Appendix

Example Dockerfile

```
FROM python:3.8-slim
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

End of Document