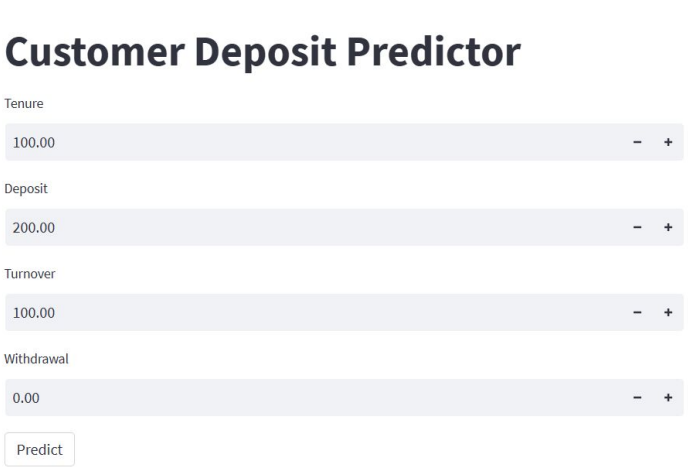
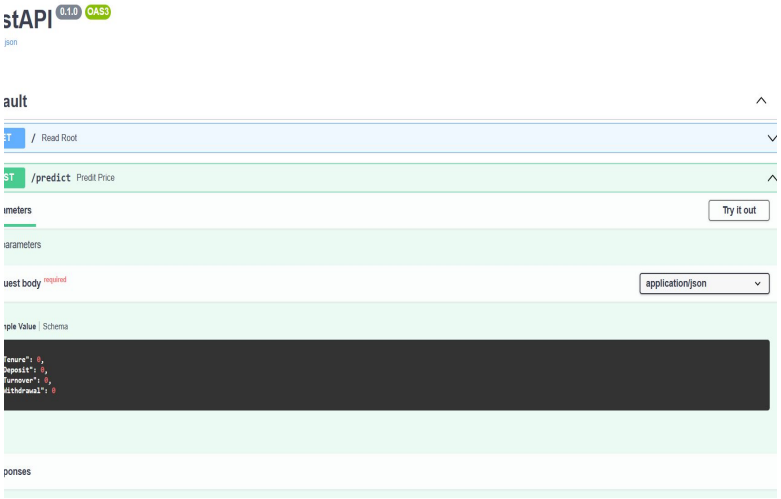
Machine Learning REST API Model

Implemented By:
Anusha

Regression Model with Backend REST API



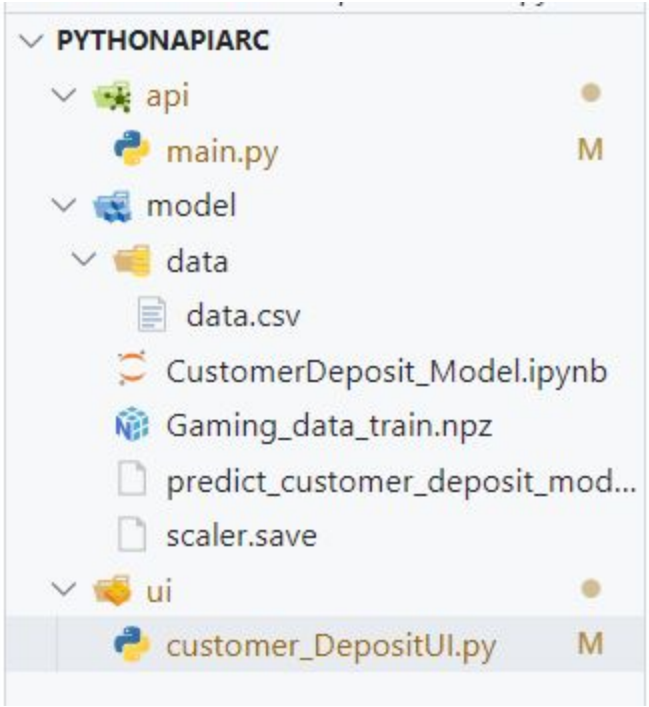
Front End



FAST API

ML Algorithm

Application Structure



Machine Learning Algorithm

Pre-processing:

Standardize
Shuffle
Split into Training, Validation and Testing

Model Architecture:

The input layer: Four nodes
The output layer: single node.
Number of hidden layers:2
Nodes in each layer: 64
Activation Function: RELU
supervised regression model with linear and nonlinear layers in the architecture.

Training:

Optimizer: Adam

Mean squared error: MSE as the loss function.

Early stopping mechanism

Evaluation:

Mean squared error (MSE) to measure the prediction error.

Plot actual vs. predicted values to visualize the model's performance.

Post-processing:

Predict the total deposit over the next 30 days for new customers.

Inverse-transform the predicted value back to the original scale using the same scaler used in the pre-processing step.

Develop REST API

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "tenure": 10,
  "deposit": 10,
  "turnover": 30,
  "withdrawal": 0
}'
```

Request URL

```
http://127.0.0.1:8000/predict
```

Server response

Code	Details
200	Response body

```
{
  "predictions": 360.044921875
}
```

Download

Response headers

```
content-length: 29
content-type: application/json
date: Wed,10 May 2023 12:04:22 GMT
server: uvicorn
```

Responses

Code	Description	Links

Develop Front-end UI using Streamlit

MLOPS Automated continuous CI/CD pipelines for continuous Deployment

- Training the Supervised Regression Model.
- Implement the Streamlit Web application.
- Create automated build script.
- Trigger build into K8s.
- Github Integration.
- Automated CI/CD pipelines.

Automated ML Deployment on kubernetes