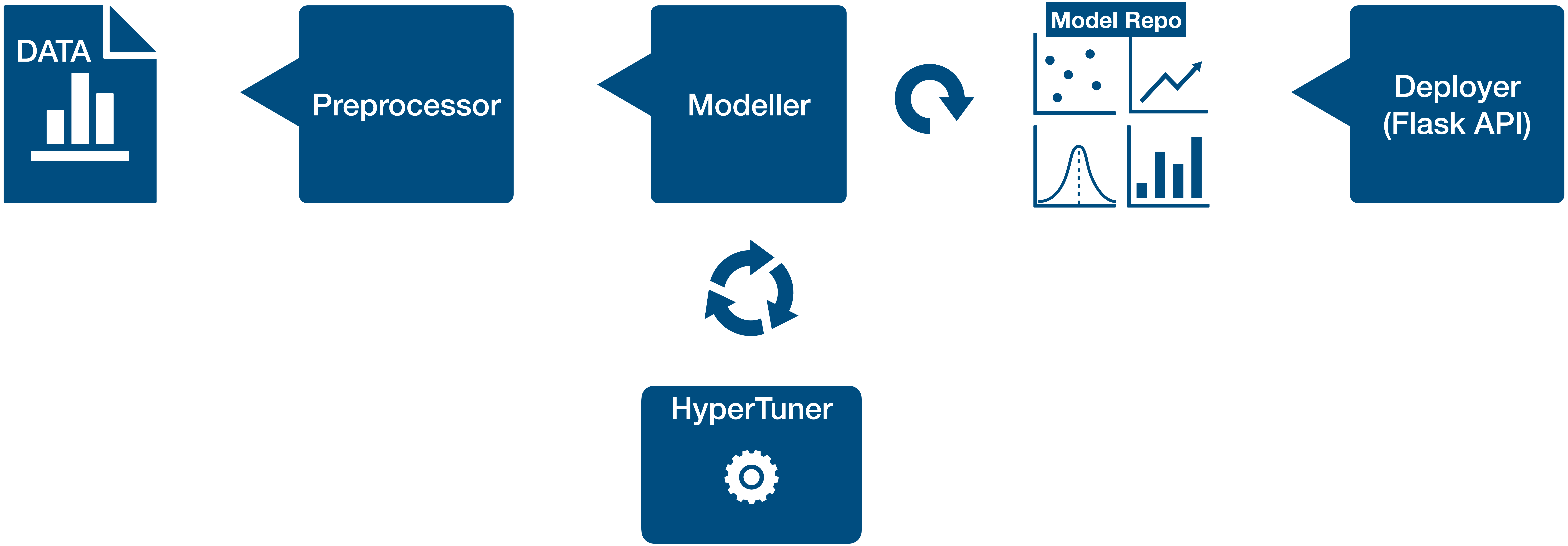
Meta Modeller for Supervised Learning

Final Semester Review

BORUSU SIVA

11-Sept-2022

Flow Diagram



DATA

Preprocessor

Modeller

Model Repo

Deployer
(Flask API)

HyperTuner

Tech Stack

- Agile Project Management Strategy is adapted for incremental updates (MVP first, then increments)
- Python with Object Oriented Programming approach for better modularity and management
- GitHub as a code Repository as a part of DevOps
- Flask Framework as a Deployment Platform as a part of DevOps
- Visual Studio Code as a IDE for quick coding and prototyping
- Anaconda Bundle with Jupypter Notebook for quick exploration and demonstrations

Mid-Sem Review - Recap

Progress Made

- Four Regression Models Implementation was Completed for MVP - Ready
- Deployment GUI and Application - Ready
- Feature Selection - Pending
- Classification Models Implementation - Pending
- Hyperopt exploration for Hyper Parameter tuning - Pending

Examiner’s Feedback

- The framework is nice and data agnostic. Good progress made in this area.
- Try to make this framework suitable for domain specific use cases.

Progress Made Post Mid-Sem Review

- Implementation is ready with Regression and Classification models, Six models each.

- Optional Feature Selection is implemented with Recursive Feature Elimination (RFECV).

- Hyper-Parameter tuning is implemented.

- Hyperopt is explored and has limitations while dealing with Random Forest, continued with Random Search with Cross Validation.

- Included two metrics each for Regression and Classification. Two separate model stores are maintained for Regression and Classification.

- Models deployment with Flask API is implemented. Made Changes to the UI with dropdown menus to mistake proof while typing model names.

- Domain Specific Machine Learning can be extended for NLP (word-net) and Computer Vision (image-net) tasks (primarily Deep Learning use cases). Traditional machine learning models are challenging to be customised for Domain Specific use cases because all the features are single numeric and encoded categorical rather represented in embeddings. Gathering Domain Specific datasets is also a major challenge in this area along with hardware accelerators.

Dataset Combinations used for Training

1. All numerical features in the dataset for Regression
2. Mix of Numerical and Categorical features in the dataset for Regression
3. All numerical features in the dataset with Binary Classification
4. Mix of Numerical and Categorical for Binary Classification
5. Mix of Numerical and Categorical variables for Multi Class Classification

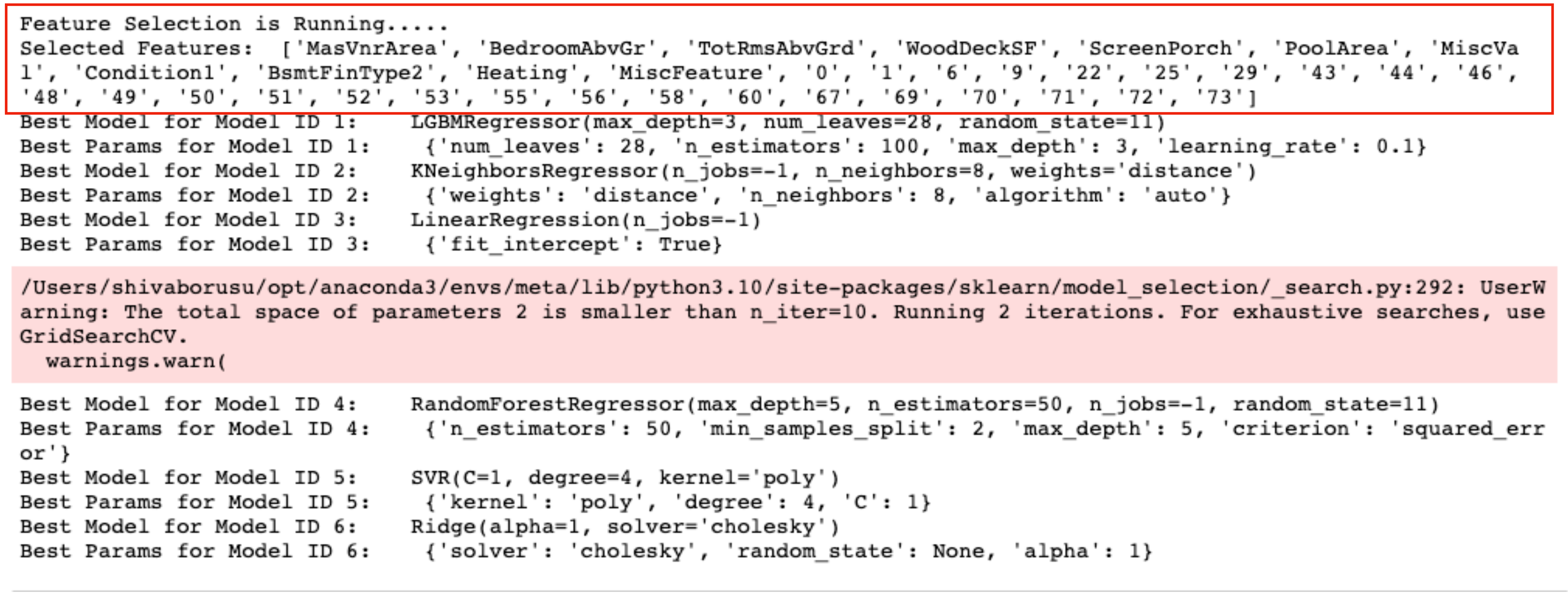
Regression Algorithms

1. Light GBM Regressor
2. K - Nearest Neighbours Regressor
3. Linear Regression
4. Random Forest Regressor
5. Support Vector Regressor (SVR)
6. Ridge (Regularised)

Classification Algorithms

1. Light GBM Classifier
2. K - Nearest Neighbours Classifier
3. Logistic Regresson
4. Random Forest Classifier
5. Support Vector Classifier (SVC)
6. Ridge Classifier

Demo Images - Model Training

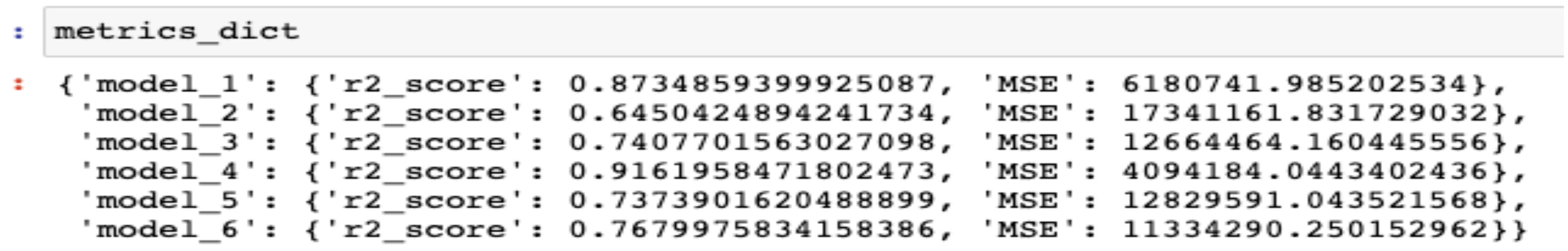


```
Feature Selection is Running.....
Selected Features:  ['MasVnrArea', 'BedroomAbvGr', 'TotRmsAbvGrd', 'WoodDeckSF', 'ScreenPorch', 'PoolArea', 'MiscVa
l', 'Condition1', 'BsmtFinType2', 'Heating', 'MiscFeature', '0', '1', '6', '9', '22', '25', '29', '43', '44', '46',
'48', '49', '50', '51', '52', '53', '55', '56', '58', '60', '67', '69', '70', '71', '72', '73']
Best Model for Model ID 1:    LGBMRegressor(max_depth=3, num_leaves=28, random_state=11)
Best Params for Model ID 1:    {'num_leaves': 28, 'n_estimators': 100, 'max_depth': 3, 'learning_rate': 0.1}
Best Model for Model ID 2:    KNeighborsRegressor(n_jobs=-1, n_neighbors=8, weights='distance')
Best Params for Model ID 2:    {'weights': 'distance', 'n_neighbors': 8, 'algorithm': 'auto'}
Best Model for Model ID 3:    LinearRegression(n_jobs=-1)
Best Params for Model ID 3:    {'fit_intercept': True}

/Users/shivaborusu/opt/anaconda3/envs/meta/lib/python3.10/site-packages/sklearn/model_selection/_search.py:292: UserW
arning: The total space of parameters 2 is smaller than n_iter=10. Running 2 iterations. For exhaustive searches, use
GridSearchCV.
  warnings.warn(

Best Model for Model ID 4:    RandomForestRegressor(max_depth=5, n_estimators=50, n_jobs=-1, random_state=11)
Best Params for Model ID 4:    {'n_estimators': 50, 'min_samples_split': 2, 'max_depth': 5, 'criterion': 'squared_err
or'}
Best Model for Model ID 5:    SVR(C=1, degree=4, kernel='poly')
Best Params for Model ID 5:    {'kernel': 'poly', 'degree': 4, 'C': 1}
Best Model for Model ID 6:    Ridge(alpha=1, solver='cholesky')
Best Params for Model ID 6:    {'solver': 'cholesky', 'random_state': None, 'alpha': 1}
```

Feature Selection is chosen



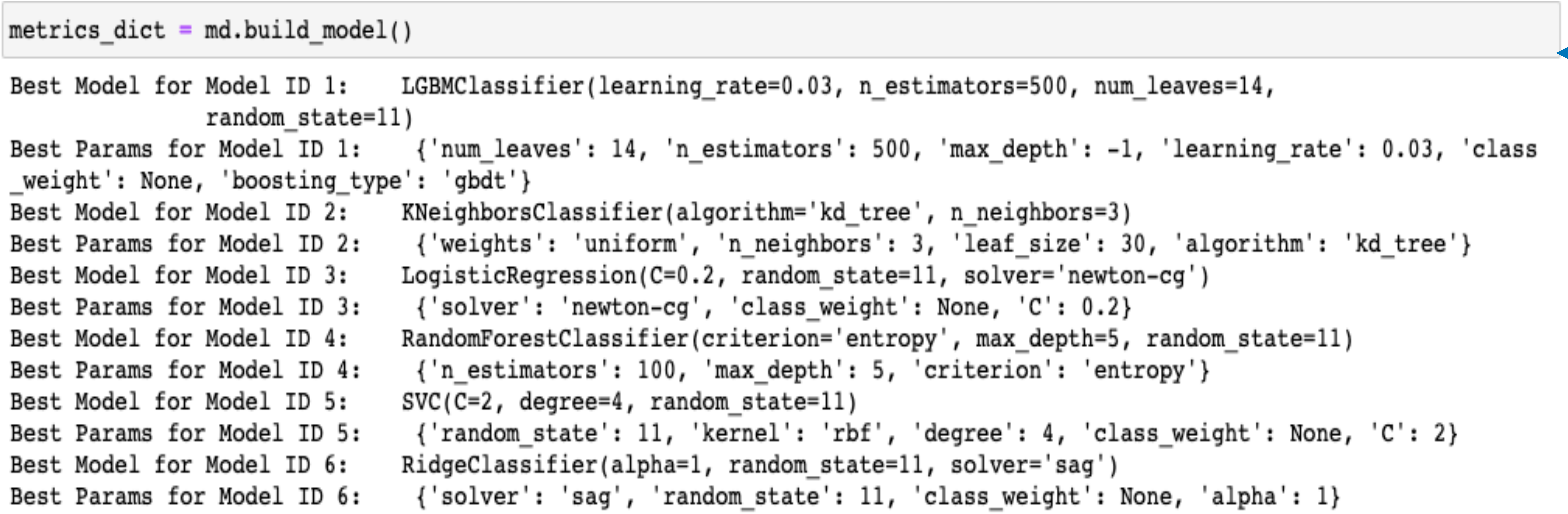
```
: metrics_dict

: {'model_1': {'r2_score': 0.8734859399925087, 'MSE': 6180741.985202534},
   'model_2': {'r2_score': 0.6450424894241734, 'MSE': 17341161.831729032},
   'model_3': {'r2_score': 0.7407701563027098, 'MSE': 12664464.160445556},
   'model_4': {'r2_score': 0.9161958471802473, 'MSE': 4094184.0443402436},
   'model_5': {'r2_score': 0.7373901620488899, 'MSE': 12829591.043521568},
   'model_6': {'r2_score': 0.7679975834158386, 'MSE': 11334290.250152962}}
```

Metrics Dictionary

Note: The warning mentioned above is due to not many tuneable hyper parameters for Linear Regression model, this warning can be suppressed but kept intentionally as a cue to the developer

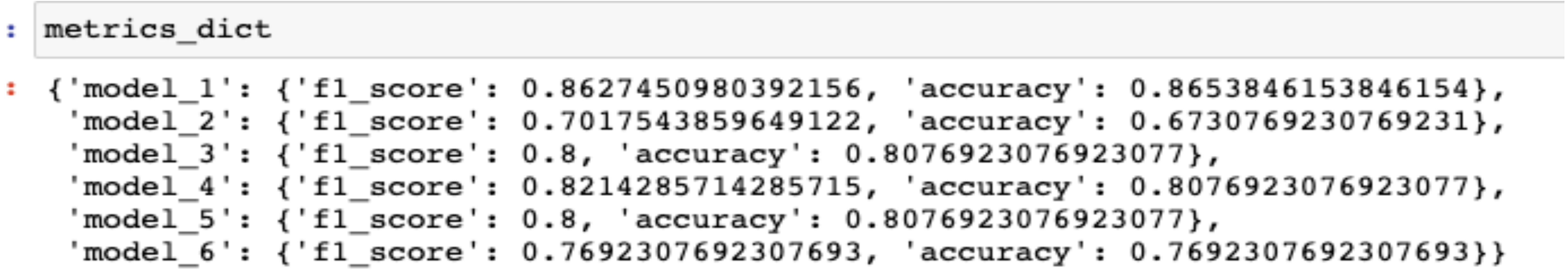
Demo Images - Model Training



```
metrics_dict = md.build_model()
```

```
Best Model for Model ID 1:    LGBMClassifier(learning_rate=0.03, n_estimators=500, num_leaves=14,
               random_state=11)
Best Params for Model ID 1:    {'num_leaves': 14, 'n_estimators': 500, 'max_depth': -1, 'learning_rate': 0.03, 'class
_weight': None, 'boosting_type': 'gbdt'}
Best Model for Model ID 2:    KNeighborsClassifier(algorithm='kd_tree', n_neighbors=3)
Best Params for Model ID 2:    {'weights': 'uniform', 'n_neighbors': 3, 'leaf_size': 30, 'algorithm': 'kd_tree'}
Best Model for Model ID 3:    LogisticRegression(C=0.2, random_state=11, solver='newton-cg')
Best Params for Model ID 3:    {'solver': 'newton-cg', 'class_weight': None, 'C': 0.2}
Best Model for Model ID 4:    RandomForestClassifier(criterion='entropy', max_depth=5, random_state=11)
Best Params for Model ID 4:    {'n_estimators': 100, 'max_depth': 5, 'criterion': 'entropy'}
Best Model for Model ID 5:    SVC(C=2, degree=4, random_state=11)
Best Params for Model ID 5:    {'random_state': 11, 'kernel': 'rbf', 'degree': 4, 'class_weight': None, 'C': 2}
Best Model for Model ID 6:    RidgeClassifier(alpha=1, random_state=11, solver='sag')
Best Params for Model ID 6:    {'solver': 'sag', 'random_state': 11, 'class_weight': None, 'alpha': 1}
```

Feature Selection is **not** chosen

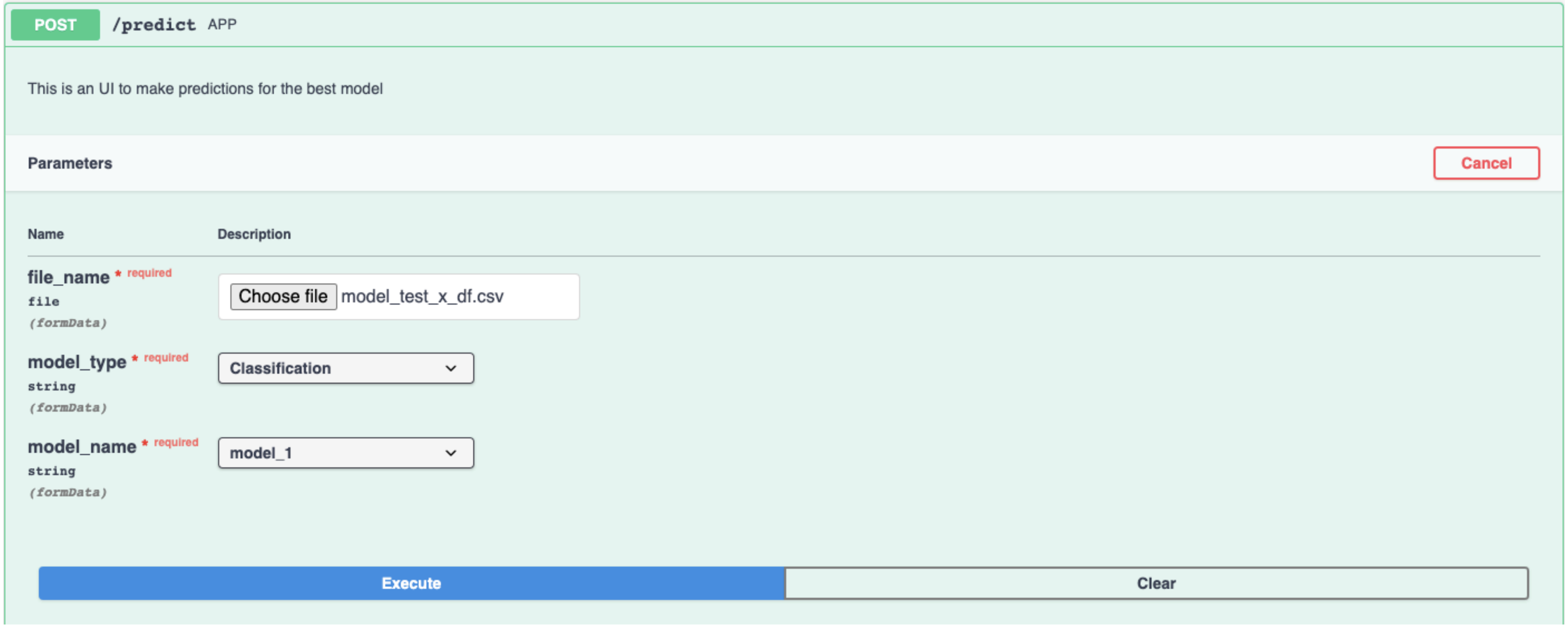


```
: metrics_dict
```

```
: {'model_1': {'f1_score': 0.8627450980392156, 'accuracy': 0.8653846153846154},
   'model_2': {'f1_score': 0.7017543859649122, 'accuracy': 0.6730769230769231},
   'model_3': {'f1_score': 0.8, 'accuracy': 0.8076923076923077},
   'model_4': {'f1_score': 0.8214285714285715, 'accuracy': 0.8076923076923077},
   'model_5': {'f1_score': 0.8, 'accuracy': 0.8076923076923077},
   'model_6': {'f1_score': 0.7692307692307693, 'accuracy': 0.7692307692307693}}
```

Metrics Dictionary

Demo Images - Deployment



POST /predict APP

This is an UI to make predictions for the best model

Parameters

Cancel

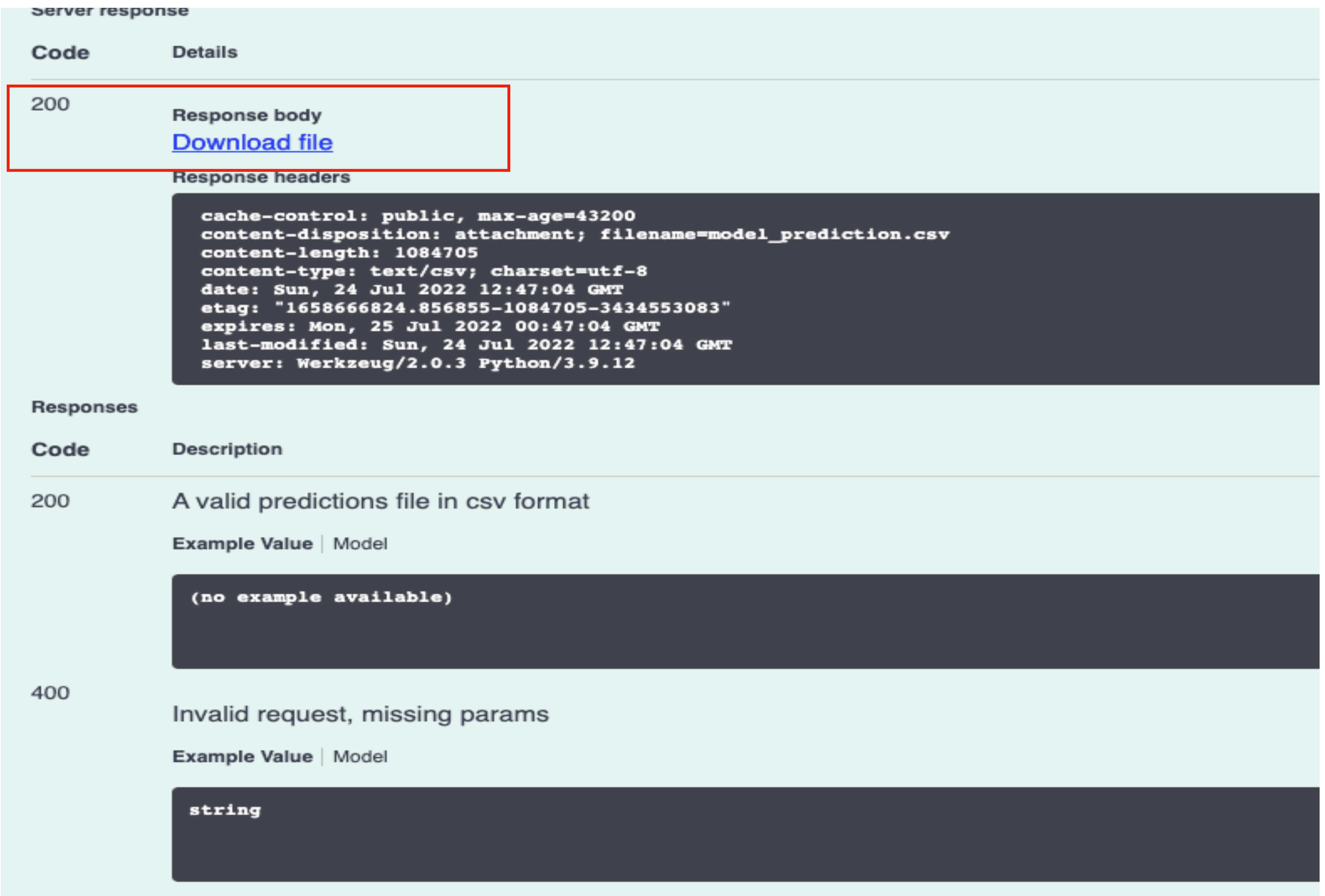
Name	Description
file_name * required file *(formData)*	Choose file model_test_x_df.csv
model_type * required string *(formData)*	Classification
model_name * required string *(formData)*	model_1

Execute

Clear

Swagger UI to upload test first and select the model

Demo Images - Deployment



Server response

Code	Details
200	**Response body** Download file **Response headers**

```
cache-control: public, max-age=43200
content-disposition: attachment; filename=model_prediction.csv
content-length: 1084705
content-type: text/csv; charset=utf-8
date: Sun, 24 Jul 2022 12:47:04 GMT
etag: "1658666824.856855-1084705-3434553083"
expires: Mon, 25 Jul 2022 00:47:04 GMT
last-modified: Sun, 24 Jul 2022 12:47:04 GMT
server: Werkzeug/2.0.3 Python/3.9.12
```

Responses

Code	Description
200	A valid predictions file in csv format **Example Value** | Model

```
(no example available)
```

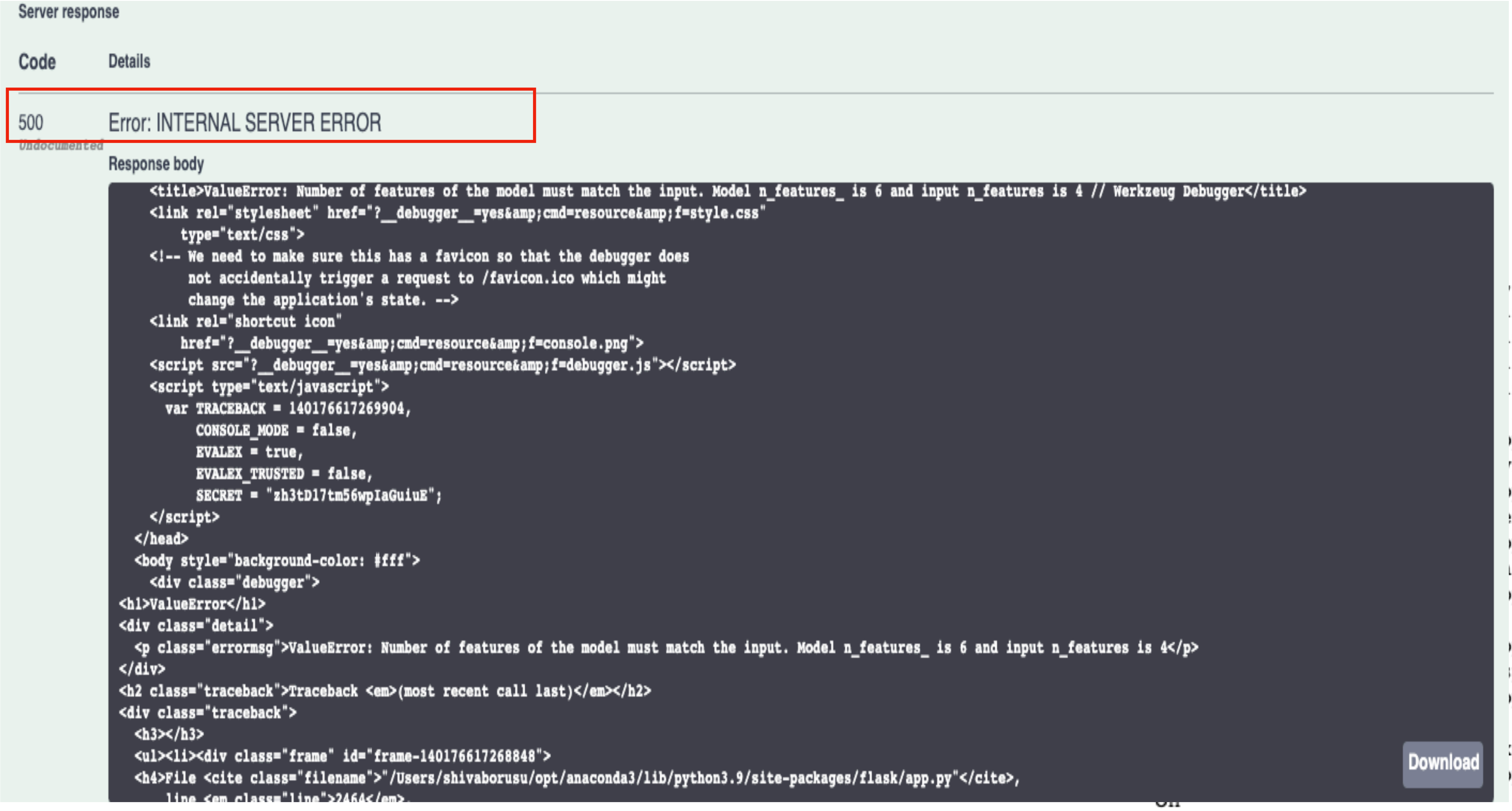
Code	Description
400	Invalid request, missing params **Example Value** | Model

```
string
```

Swagger UI to Download Predictions

Demo Images - Deployment

ERROR Screens when improper Data or Model is Chosen for Predictions



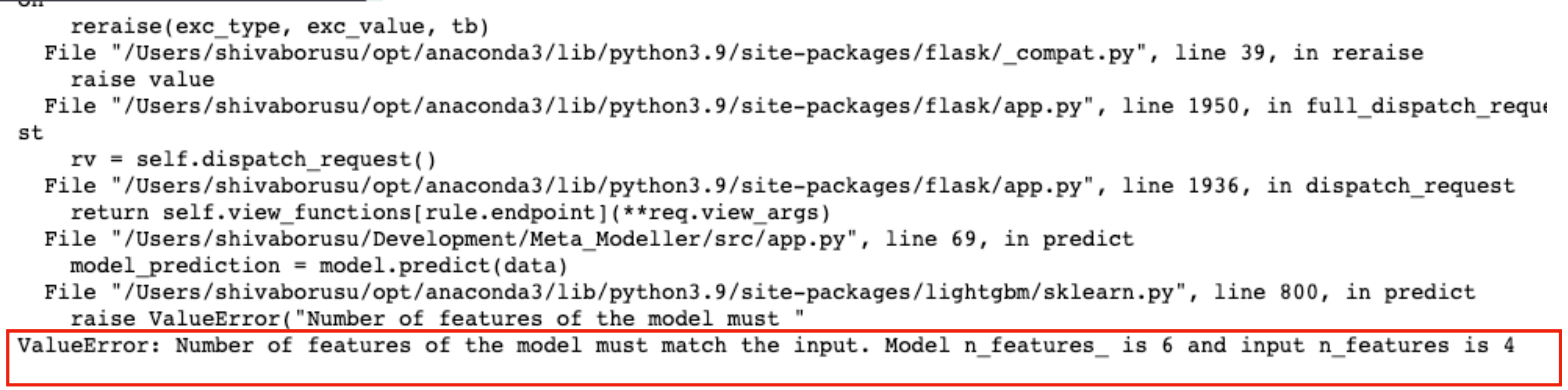
Server response

Code	Details
500 Undocumented	Error: INTERNAL SERVER ERROR

Response body

```
    <title>ValueError: Number of features of the model must match the input. Model n_features_ is 6 and input n_features is 4 // Werkzeug Debugger</title>
    <link rel="stylesheet" href="?__debugger__=yes&amp;cmd=resource&amp;f=style.css"
        type="text/css">
    <!-- We need to make sure this has a favicon so that the debugger does
         not accidentally trigger a request to /favicon.ico which might
         change the application's state. -->
    <link rel="shortcut icon"
        href="?__debugger__=yes&amp;cmd=resource&amp;f=console.png">
    <script src="?__debugger__=yes&amp;cmd=resource&amp;f=debugger.js"></script>
    <script type="text/javascript">
      var TRACEBACK = 140176617269904,
          CONSOLE_MODE = false,
          EVALEX = true,
          EVALEX_TRUSTED = false,
          SECRET = "zh3tDl7tm56wpIaGuiuE";
    </script>
  </head>
  <body style="background-color: #fff">
    <div class="debugger">
<h1>ValueError</h1>
<div class="detail">
  <p class="errormsg">ValueError: Number of features of the model must match the input. Model n_features_ is 6 and input n_features is 4</p>
</div>
<h2 class="traceback">Traceback <em>(most recent call last)</em></h2>
<div class="traceback">
  <h3></h3>
  <ul><li><div class="frame" id="frame-140176617268848">
  <h4>File <cite class="filename">"/Users/shivaborusu/opt/anaconda3/lib/python3.9/site-packages/flask/app.py"</cite>,
      line <em class="line">2464</em>,
```

Download



```
                0:41:10] "GET /apidocs/ HTTP/1.1" 200 -
                0:41:10] "GET /apispec_1.json HTTP/1.1" 200 -
                0:41:29] "POST /predict HTTP/1.1" 200 -
                0:41:48] "POST /predict HTTP/1.1" 500 -
                 last):
                pt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 2464, in __call__
                iron, start_response)
                pt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 2450, in wsgi_app
                xception(e)
                pt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1867, in handle_exception
                lue, tb)
                pt/anaconda3/lib/python3.9/site-packages/flask/_compat.py", line 39, in reraise

                pt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 2447, in wsgi_app
                patch_request()
                pt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1952, in full_dispatch_reque

                ception(e)
                pt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1821, in handle_user_except
on
    reraise(exc_type, exc_value, tb)
  File "/Users/shivaborusu/opt/anaconda3/lib/python3.9/site-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/Users/shivaborusu/opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1950, in full_dispatch_reque
st
    rv = self.dispatch_request()
  File "/Users/shivaborusu/opt/anaconda3/lib/python3.9/site-packages/flask/app.py", line 1936, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/Users/shivaborusu/Development/Meta_Modeller/src/app.py", line 69, in predict
    model_prediction = model.predict(data)
  File "/Users/shivaborusu/opt/anaconda3/lib/python3.9/site-packages/lightgbm/sklearn.py", line 800, in predict
    raise ValueError("Number of features of the model must "
ValueError: Number of features of the model must match the input. Model n_features_ is 6 and input n_features is 4
```

References

Building Domain-Specific Machine Learning Workflows: A Conceptual Framework for the State-of-the-Practice

https://arxiv.org/abs/2203.08638

Best Practices for Creating Domain-Specific AI Models

https://www.kdnuggets.com/2022/07/best-practices-creating-domainspecific-ai-models.html

Thank You…!