Assignment :3

B kanaka durga

Title: kidney stone prediction


```
{
 "cells":
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "e7f01297",
   "metadata": {
    "_cell_guid": "b1076dfc-b9ad-4769-8c92-a6c4dae69d19",
    "_uuid": "8f2839f25d086af736a60e9eeb907d3b93b6e0e5",
    "execution": {
     "iopub.execute_input": "2024-02-27T09:07:17.602215Z",
     "iopub.status.busy": "2024-02-27T09:07:17.601420Z",
     "iopub.status.idle": "2024-02-27T09:07:18.645546Z",
     "shell.execute_reply": "2024-02-27T09:07:18.644642Z"
    },
    "papermill": {
     "duration": 1.052285,
     "end_time": "2024-02-27T09:07:18.648803",
     "exception": false,
     "start_time": "2024-02-27T09:07:17.596518",
```

```
   "status": "completed"
  },
  "tags": []
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "/kaggle/input/playground-series-s3e12/sample_submission.csv\n",
    "/kaggle/input/playground-series-s3e12/train.csv\n",
    "/kaggle/input/playground-series-s3e12/test.csv\n"
   ]
  }
 ],
 "source": [
  "# This Python 3 environment comes with many helpful analytics libraries installed\n",
  "# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python\n",
  "# For example, here's several helpful packages to load\n",
  "\n",
  "import numpy as np # linear algebra\n",
  "import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)\n",
  "\n",
  "# Input data files are available in the read-only \"../input/\" directory\n",
  "# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory\n",
```

```
   "\n",

   "import os\n",

   "for dirname, _, filenames in os.walk('/kaggle/input'):\n",

   "    for filename in filenames:\n",

   "        print(os.path.join(dirname, filename))\n",

   "\n",

   "# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using \"Save & Run All\" \n",

   "# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session"

  ]

 },

 {

  "cell_type": "code",

  "execution_count": 2,

  "id": "2a178bb1",

  "metadata": {

   "execution": {

    "iopub.execute_input": "2024-02-27T09:07:18.656932Z",

    "iopub.status.busy": "2024-02-27T09:07:18.656091Z",

    "iopub.status.idle": "2024-02-27T09:07:21.031584Z",

    "shell.execute_reply": "2024-02-27T09:07:21.030378Z"

   },

   "papermill": {

    "duration": 2.385295,

    "end_time": "2024-02-27T09:07:21.036967",
```

```
  "exception": false,
  "start_time": "2024-02-27T09:07:18.651672",
  "status": "completed"
 },
 "tags": []
},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "Validation ROC AUC Score: 0.8380116959064328\n"
  ]
 }
],
"source": [
 "import pandas as pd\n",
 "from sklearn.model_selection import train_test_split\n",
 "from sklearn.ensemble import RandomForestClassifier\n",
 "from sklearn.metrics import roc_auc_score\n",
 "\n",
 "# Load the training and test data\n",
 "train_data = pd.read_csv(\"/kaggle/input/playground-series-s3e12/train.csv\")\n",
 "test_data = pd.read_csv(\"/kaggle/input/playground-series-s3e12/test.csv\")\n",
 "\n",
```

```
"# Separate features and target variable\n",
"X_train = train_data.drop(columns=[\"target\"])\n",
"y_train = train_data[\"target\"]\n",
"\n",
"# Perform train-test split for model evaluation\n",
"X_train_split, X_val, y_train_split, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)\n",
"\n",
"# Initialize the classifier\n",
"clf = RandomForestClassifier(n_estimators=100, random_state=42)\n",
"\n",
"# Train the classifier\n",
"clf.fit(X_train_split, y_train_split)\n",
"\n",
"# Make predictions on the validation set\n",
"y_val_pred_proba = clf.predict_proba(X_val)[:, 1]\n",
"\n",
"# Calculate the ROC AUC score\n",
"roc_auc = roc_auc_score(y_val, y_val_pred_proba)\n",
"print(\"Validation ROC AUC Score:\", roc_auc)\n",
"\n",
"# Make predictions on the test data\n",
"test_predictions = clf.predict_proba(test_data)[:, 1]\n",
"\n",
"# Create a submission DataFrame\n",
"submission_df = pd.DataFrame({\n",
```

```
   "    \"id\": test_data[\"id\"],\n",
   "    \"target\": test_predictions\n",
   "})\n",
   "\n",
   "# Save the submission file\n",
   "submission_df.to_csv(\"/kaggle/working/submission.csv\", index=False)\n"
  ]
 }
],
"metadata": {
 "kaggle": {
  "accelerator": "none",
  "dataSources": [
   {
    "databundleVersionId": 5220863,
    "sourceId": 49200,
    "sourceType": "competition"
   }
  ],
  "dockerImageVersionId": 30646,
  "isGpuEnabled": false,
  "isInternetEnabled": true,
  "language": "python",
  "sourceType": "notebook"
 },
```

```
"kernelspec": {
 "display_name": "Python 3",
 "language": "python",
 "name": "python3"
},
"language_info": {
 "codemirror_mode": {
  "name": "ipython",
  "version": 3
 },
 "file_extension": ".py",
 "mimetype": "text/x-python",
 "name": "python",
 "nbconvert_exporter": "python",
 "pygments_lexer": "ipython3",
 "version": "3.10.13"
},
"papermill": {
 "default_parameters": {},
 "duration": 7.999535,
 "end_time": "2024-02-27T09:07:21.867334",
 "environment_variables": {},
 "exception": null,
 "input_path": "__notebook__.ipynb",
 "output_path": "__notebook__.ipynb",
```

```
  "parameters": {},
  "start_time": "2024-02-27T09:07:13.867799",
  "version": "2.5.0"
 }
},
"nbformat": 4,
"nbformat_minor": 5
}
```

Files

* train.csv - the training dataset; target is the likelihood of a kidney stone being present

* test.csv - the test dataset; your objective is to predict the probability of target

* sample_submission.csv - a sample submission file in the correct format