```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "name": "Arsha DJ. ipynb",
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "metadata": {
        "id": "NKoUoSmo7c2L"
      },
      "source": [
        " \n",
        "from keras. datasets import mnist \n",
        "data=mnist. load_data()"
      ],
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "ajWoqq927ekL"
      },
      "source": [
        " \n",
        "((X_train,  y_train),  (X_test,  y_test)) =\n",
        "data"
      ],
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "x7sl4ov37g5l"
      },
      "source": [
        " \n",
        "X_train = X_train. reshape((X_train.  shape [0],  28*28)). astype('float32')\n",
        "X_test = X_test.  reshape( (x_test. shape [0],  28*28)). astype('float32')"
      ],
      "execution_count": null,
      "outputs": []
    },
{
   "cell_type": "code",
   "metadata": {
      "id": "rMcg2DXX7jo2"
   },
   "source": [
      " \n",
      "X_train = X_train / 255\n",
      "X_test = X_test / 255"
   ],
   "execution_count": null,
   "outputs": []
},
{
   "cell_type": "code",
   "metadata": {
      "id": "aTEkNR_g7mlp"
   },
   "source": [
      " \n",
      "from keras. utils import np_utils\n",
      "print(y_test.  shape)\n",
      "y_train = np_utils. to_categorically_train)\n",
      "y_test = np_utils. to_categorically(y_test)\n",
      "num_classes = y_test. shape [1]\n",
      "print(y_test. shape)"
   ],
   "execution_count": null,
   "outputs": []
},
{
   "cell_type": "code",
   "metadata": {
      "id": "otntZ1O57oX3"
   },
   "source": [
      " \n",
      "from keras. models import Sequential\n",
      "from keras. layers import Dense\n",
      "from keras. layers import Dropout"
   ],
   "execution_count": null,
   "outputs": []
},
{
   "cell_type": "code",
   "metadata": {
      "id": "TUyluL8C7rfC"
   },
   "source": [
      " \n",
      "model = Sequential( )\n",
      "model. add(Densel(64,    input_dim =    28*28,  activation='relu'))\n",
      "model. add( Dense(64,  activation='relu'))\n",
      "model. add( Dense(10,  activation='softmax'))"
   ],
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "metadata": {
      "id": "foqwk6ds7uFz"
    },
    "source": [
      " \n",
      "model. compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy"
    ],
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "metadata": {
      "id": "cf7_1bTL72WJ"
    },
    "source": [
      " model. summary( )"
    ],
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "metadata": {
      "id": "wH01vlL870Kx"
    },
    "source": [
      " model. fit(X_train, y_train, validation_split=0. 2 , epochs=10, batch_size=100)"
    ],
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "metadata": {
      "id": "KyeqK-sf7zMC"
    },
    "source": [
      " \n",
      "scores = model. evaluate(x_test, y_test)\n",
      "print(scores) # returns loss and accuracy"
    ],
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "metadata": {
      "id": "i_hFTm4R7_Lk"
    },
    "source": [
        " \n",
        "predictions = model. predict(x_test)\n",
        "# Convert prediction to labels\n",
        "label=[]\n",
        "for pred in predictions:\n",
        "max_val = max(pred)\n",
        "for (i, p) in enumerate(pred):"
      ],
      "execution_count": null,
      "outputs": []
    }
  ]
}
```