Topic	Model View Controller	
Class Description	**Students learn to create a prediction model which takes data from an API and makes predictions.**	
Class	**C125**	
Class time	**45 mins**	
Goal	● Create an API with the post request. ● Create a classifier model. ● Send data through the API to the classifier to make predictions.	
Resources Required	● Teacher Resources ○ Postman Software ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ Postman Software ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 min** **5 min**

CONTEXT

- **Explore the technique of sending data to the classifier through an API**

Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	Hi <Student name>. How are you doing today? Wasn't last class fun? Can you tell me	**ESR:** We learned about the flask framework .

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	what were the new things that you learned?	We use it to create an API. We tested our API on the postman software.
	Perfect!! So we got to know how we can create an API. And we also saw how we can add data to an API using a POST request. In the last few classes we have created multiple prediction models and have provided them data in various ways. Today we are going to do one more such thing where we pass the data through the API that we create and then see how well our prediction model performs. Sounds exciting?	**ESR:** Yes!!
	Let's get started then	
Teacher Initiates Screen Share		
CHALLENGE • **Create the classifier model which takes data from the API**		
Step 2: Teacher-led Activity (15 min)	*<Teacher opens the VS Code editor>* Can you tell me what are the steps we follow while creating a prediction model?	**ESR:** 1.We first import all the dependencies/ libraries. 2. Then we get the data to train and test the model. 3. Then we split the data to train and test the model. 4. Then we scale the data to make it of equal size. 5. Then we fit the data into

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

		the logistic regression model. 6.Then we make predictions.
	Yes! Alright so let's start with our prediction model to which we'll be providing the data through the API that we create.	-
	So let's start. What's the first step? Yes! *<Teacher creates a classifier.py file and codes to import all the libraries that are needed for the model.>* Code: **import numpy as np** **import pandas as pd** **from sklearn.datasets import fetch_openml** **from sklearn.model_selection import train_test_split** **from sklearn.linear_model import LogisticRegression** **from PIL import Image** **import PIL.ImageOps**	**ESR:** To Import all the libraries.

```
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from PIL import Image
import PIL.ImageOps
```

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Now we have all the libraries imported, now it's time to import the data that we'll be using to train and test the model. A class before we learned to take the data from the data set provided by the machine learning dataset provided by the sklearn . So we are going to use the digits image data that we used before in one of our previous classes. *<Teacher codes to import the data.>* Code: **X, y = fetch_openml('mnist_784', version=1, return_X_y=True)** Here we are using the fetch_openml function to get data, the name of the data set is 'mnist_784, version =1, and getting the x and y values of it.	*The student helps the teacher with the code.*

```
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)
```

	What's the next step? *<Teacher codes to split the data to train and test the model.>* Code: **X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=9, train_size=7500, test_size=2500)** What is the ideal training and testing	**ESR:** To split the data to train and test the model. **ESR:** 75% for training ,25% for testing.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	size of the data ? Yes.	

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=9,
train_size=7500, test_size=2500)
```

	Now we need to scale the data to make sure that the data points in X and Y are equal so we'll divide them using 255 which is the maximum pixel of the image. *<Teacher codes to scale the data.>* Code: **X_train_scaled = X_train/255.0** **X_test_scaled = X_test/255.0**	*The student helps the teacher with the code.*

```
X_train_scaled = X_train/255.0
X_test_scaled = X_test/255.0
```

	After scaling the data we have to fit it inside our model so that it can give output with maximum accuracy. *<Teacher codes to fit the data in the logistic regression model.>* Code: **clf = LogisticRegression(solver='saga', multi_class='multinomial').fit(X_train_scaled, y_train)**	*The student helps teacher with the code.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
clf = LogisticRegression(solver='saga',
multi_class='multinomial').fit(X_train_scaled, y_train)
```

	Now, we have our classifier ready. Using this classifier, if we have an image, can we predict the digit mentioned on the image? If we remember what we did earlier, we were using CV2 and using CV2, we were using our device's camera and capturing each frame. Now, each frame was an image where we were doing some processing and then predicting the value from it. Let's create a function to do that. We'll call it get_prediction which will take the image as the parameter and make a prediction. This function will take the image and convert it into a scalar quantity and then make it grey so that the colors don't affect the prediction. And then resize it into 28 by 28 scales. Then using the percentile function get the minimum pixel and and then using the clip function give each image a number. And then get the maximum pixel and make an array of this. And then create a test sample of it and make predictions based on that sample. And then finally return the test prediction.	**ESR:** Yes.

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Code:

```
def get_prediction(image):
    im_pil = Image.open(image)
    image_bw = im_pil.convert('L')
    image_bw_resized =
image_bw.resize((28,28),
Image.ANTIALIAS)
    pixel_filter = 20
    min_pixel =
np.percentile(image_bw_resized,
pixel_filter)

image_bw_resized_inverted_scale
d =
np.clip(image_bw_resized-min_pix
el, 0, 255)
    max_pixel =
np.max(image_bw_resized)

image_bw_resized_inverted_scale
d =
np.asarray(image_bw_resized_inve
rted_scaled)/max_pixel
    test_sample =
np.array(image_bw_resized_invert
ed_scaled).reshape(1,784)
    test_pred =
clf.predict(test_sample)
    return test_pred[0]
```

```
def get_prediction(image):
    im_pil = Image.open(image)
    image_bw = im_pil.convert('L')
    image_bw_resized = image_bw.resize((28,28), Image.ANTIALIAS)
    pixel_filter = 20
    min_pixel = np.percentile(image_bw_resized, pixel_filter)
```

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
    image_bw_resized_inverted_scaled = np.clip(image_bw_resized-min_pixel,
0, 255)
    max_pixel = np.max(image_bw_resized)
    image_bw_resized_inverted_scaled =
np.asarray(image_bw_resized_inverted_scaled)/max_pixel
    test_sample = np.array(image_bw_resized_inverted_scaled).reshape(1,784)
    test_pred = clf.predict(test_sample)
    return test_pred[0]
```

	Now that our classifier is complete, how can we pass an image into this function that we just created? Can you try writing the API and use the classifier function in it?	**ESR:** We can create an API which accepts an image. **ESR:** Yes.
	Let's get started then	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

ACTIVITY

- **Create an API with POST request**
- **Send data through the API and test the classifier model**

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Step 3: Student-Led Activity (15 min)	*<Teacher helps student to open the code editor and create a classifier.py file and write the get_prediction function.>*	*Student opens the code editor, creates a classifier.py and writes code for the get_prediction function.*
	Alright now we'll start creating our API. First we'll import Flask, requests to make request on the API and jsonify to convert data into json. *<Teacher helps student to import Flask ,requests and jsonify from flask.>* Code: **from flask import Flask, jsonify, request**	*Student codes to import Flask, requests and jsonify from flask.*
	As we know that every python program is a module and can be used in other programs too. So here we'll use the classifier function that we created before in this code. So we are going to import the get_prediction function from the classifier file. *<Teacher helps student with the*	*Students code to import the get_prediction function.*

```
from flask import Flask, jsonify, request
```

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	code.> Code: **from classifier import get_prediction**	

```
from classifier import  get_prediction
```

	We'll now give the name of the app to the flask constructor so that it'll know where to look for the dependent files. Code: **app = Flask(__name__)**	*The student codes to set the name of the app in the flask constructor.*

```
app = Flask(__name__)
```

	Now we'll start writing our route. We need a post request to send the image to the prediction model and our route name would be 'predict-digit'. Code: **@app.route("/predict-digit", methods=["POST"])**	*Student codes to write the route with a post request.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
@app.route("/predict-digit", methods=["POST"])
```

	Inside this route we'll write a function called predict_data(). We'll use the request function to get the files from the digit key. We'll also use the get_prediction function here and return the prediction in the json format with the status code of 200. Code: **def predict_data():** **image = request.files.get("digit")** **prediction = get_prediction(image)** **return jsonify({** **"prediction": prediction** **}), 200**	*The student codes to write the predict_data function.*

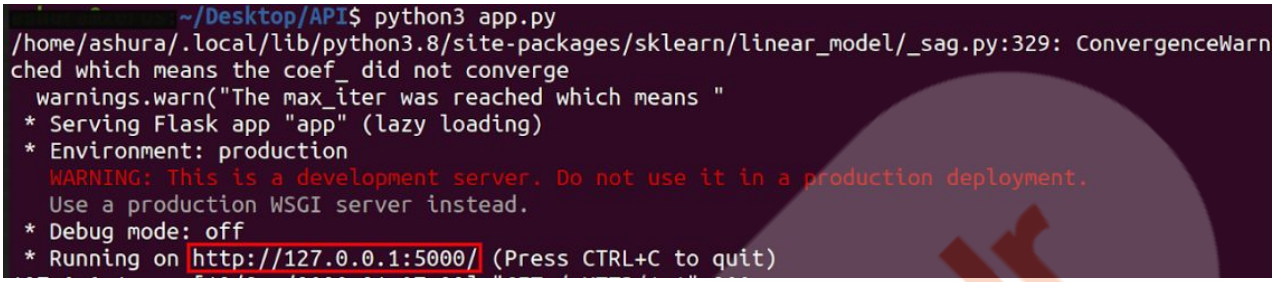
```
def predict_data():
 image = request.files.get("digit")
 prediction = get_prediction(image)
 return jsonify({
   "prediction": prediction
 }), 200
```

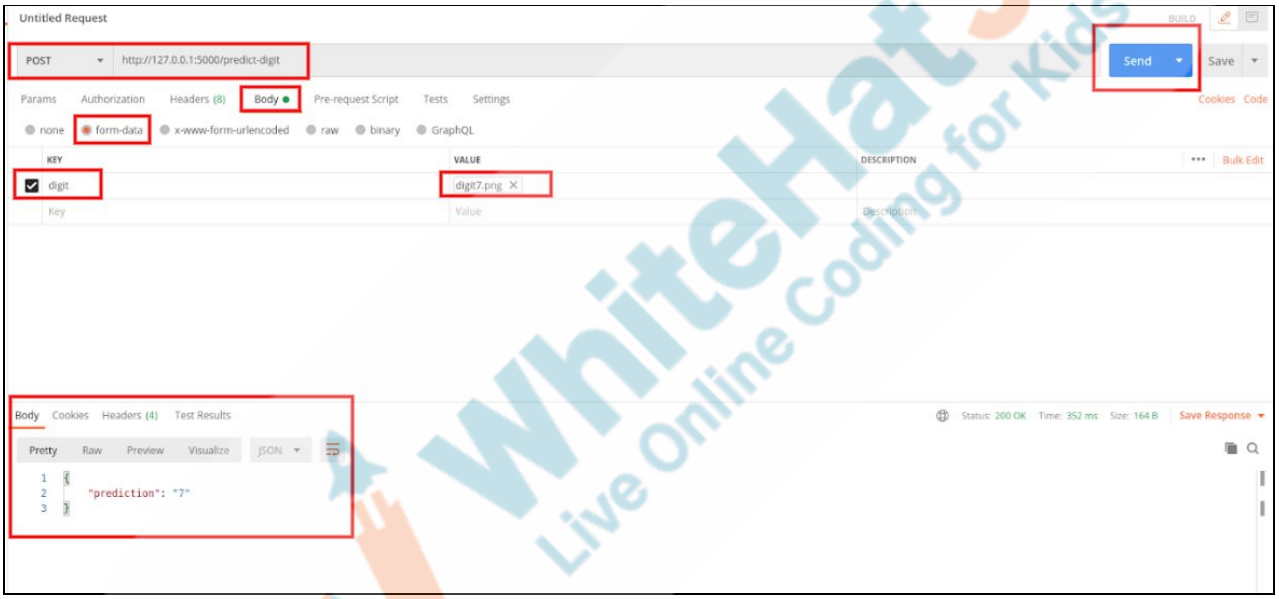
© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Then we'll use the' if __name__ == "main" ' block to prevent (certain) code from being run when the module is imported. *<Teacher helps student with the code.>* Code: **if __name__ == "__main__":** **app.run(debug=True)** We write debug=True so that the server will update every time you save the code.	*Student codes to write if__name__==main block.*
``if __name__ == "__main__": app.run(debug=True)```		
	Now we'll run the code, copy the port on which the code is running and test the API in the postman. *<Teacher asks student to open the postman.>* Now make a post request on the route that we created which would be "http://127.0.0.1:5000/predict-digit". We have to make the request in the "form-data" as we'll be giving image files to the API. In the place of "key" we have to write "digit" and type as file. As its value	*Student opens the postman and tests the API and the prediction model.* *The student uses the images given in student activity 1 to test the prediction.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	we'll be using the digit images files. And then make a request. To see the prediction.	





Teacher Guides Student to Stop Screen Share

FEEDBACK

- **Appreciate the student for their efforts**
- **Identify 2 strengths and 1 area of progress for the student**

Step 4: Wrap-Up (5 min)	How did you find today's class?	*Student talks about his/her experience of today's class.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	So now we know how we can send data through an API to test the prediction model. Before we end the class, can you tell me why we created a classification model in a separate file instead of doing it in the api file itself? Also, why did we make a separate function to process the image and predict the outcome, instead of doing it in the API?	**ESR:** The Classifier takes time to learn from the training data that we provide. If we build it in the API itself, then it will take a lot of time to give the response. **ESR:** Flask APIs, to optimize the performance speed, caches the APIs so that whenever an API call is made, it can return the data much faster. If we did this in the API itself, it will return the same output for different images that we send to the server.
	What we did today is known as the MVC Architecture. MVC stands for Model View Controller .There are also other types of architectures like MVVM,MVP, MVA etc. Keeping in mind that the API caches the data so that it can give a response faster, it is ideal to keep your processing code separately from your API.	

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	Model is generally referred to the Database, if any. Here, we are not using any Databases. **View** is referred to the API, which accepts a request and returns a response. **Controller** is the part that does all the heavy work, i.e, data processing, building classifier, etc. Hence, it is known as the MVC Architecture!	
	In our next class we'll do something much more amazing. Are you up for it? Alright then, I'll see you in next class.	
Teacher Clicks 		
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: ● What happened today? - Describe what happened - Code I wrote ● How did I feel after the class?	*The student uses the markdown editor to write her/his reflection in a reflection journal.*

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

	• What have I learned about programming and developing games? • What aspects of the class helped me? What did I find difficult?	

Activity	Activity Name	Links
Teacher Activity 1	Reference code	https://github.com/whitehatjr/C125/tree/master/API
Student Activity 1	Images for testing	https://github.com/whitehatjr/C125/tree/master/API/images

© 2020 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.