**GetEventPrediction API Testing**

A S, Sarin Kumar (2021)

**Contents**

[Testing scenario – Fraud transaction from the test dataset is appeared to be fraud during testing. Rule outcome is block 2](#_Toc111627798)

[Testing scenario – Legitimate transaction from the test dataset. Rule outcome is Approved 4](#_Toc111627799)

[Testing scenario - Small Amount Followed by Big Amount 5](#_Toc111627800)

[Testing Scenario- Transactions in another, distant place 7](#_Toc111627801)

[Testing scenario - large transactions 11](#_Toc111627802)

[Testing Scenario- Multiple transactions in a short period of time 13](#_Toc111627803)

# **Testing scenario – Fraud transaction from the test dataset is appeared to be fraud during testing. Rule outcome is block**

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32910911f74289ea1b895fd46405aa8957',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '261-68-3049'

},

],

eventTimestamp='2022-02-26T10:32:35Z',

eventVariables={

'customer\_email': 'christineshepard@tate-fields.com',

'order\_price': '323.49',

'billing\_city':'Fontanelle',

'product\_category':'grocery\_pos',

'billing\_street':'979 Robin Manor Suite 057',

'user\_agent':'Mozilla/5.0 (Android 5.1; Mobile; rv:8.0) Gecko/8.0 Firefox/8.0',

'ip\_address':'120.250.35.224',

'card\_bin':'301931'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 1000.0}}], 'ruleResults': [{'ruleId': 'decline\_rule', 'outcomes': ['block']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '18177065-d0f3-49ac-aee9-48ee93f84310', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 21:44:47 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '274', 'connection': 'keep-alive', 'x-amzn-requestid': '18177065-d0f3-49ac-aee9-48ee93f84310'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32901a6d8b13eb901d379bd3fdd92c8fa7',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '619-16-3949'

},

],

eventTimestamp='2021-12-21T05:34:41Z',

eventVariables={

'customer\_email': 'brandi56@ochoa-manning.com',

'order\_price': '265.35',

'billing\_city':'Westwego',

'product\_category':'grocery\_pos',

'billing\_street':'913 Sara Locks',

'user\_agent':'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 5.1; Trident/5.1)',

'ip\_address':'118.101.78.20',

'card\_bin':'658633'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 987.0}}], 'ruleResults': [{'ruleId': 'decline\_rule', 'outcomes': ['block']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'acbe435c-5c5a-48b6-913a-557b458b12c2', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 21:55:20 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '273', 'connection': 'keep-alive', 'x-amzn-requestid': 'acbe435c-5c5a-48b6-913a-557b458b12c2'}, 'RetryAttempts': 0}}

# **Testing scenario – Legitimate transaction from the test dataset. Rule outcome is Approved**

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32a1232528f4a8c57f7c991e91d2aa7b16',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '174-88-0351'

},

],

eventTimestamp='2021-12-11T14:46:25Z',

eventVariables={

'customer\_email': 'iflores@hotmail.com',

'order\_price': '105.44',

'billing\_city':'Ossian',

'product\_category':'grocery\_pos',

'billing\_street':'4529 Brian Trail Suite 869',

'user\_agent':'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.2; Trident/5.0)',

'ip\_address':'195.181.159.94',

'card\_bin':'300361'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 461.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'd680b7f4-0f84-4220-b86d-fbe5d2405dd6', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 22:03:34 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': 'd680b7f4-0f84-4220-b86d-fbe5d2405dd6'}, 'RetryAttempts': 0}}

# **Testing scenario - Small Amount Followed by Big Amount**

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='323cca7231ff70e0e5d20ece9a54413qb4',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '523-53-7927'

},

],

eventTimestamp='2022-01-10T12:02:20Z',

eventVariables={

'customer\_email': 'zimmermannicole@williams.com',

'order\_price': '0.5',

'billing\_city':'Valier',

'product\_category':'health\_fitness',

'billing\_street':'69624 Erika Court',

'user\_agent':'Mozilla/5.0 (X11; Linux i686) AppleWebKit/532.1 (KHTML, like Gecko) Chrome/39.0.827.0 Safari/532.1',

'ip\_address':'183.83.29.194',

'card\_bin':'440955'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 188.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '5a2fb2eb-5a95-431f-8ea1-661f2a43acee', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 19:52:15 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': '5a2fb2eb-5a95-431f-8ea1-661f2a43acee'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='323cca7231ff70e0e4520ece9a54413qb4',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '523-53-7927'

},

],

eventTimestamp='2022-01-10T12:02:20Z',

eventVariables={

'customer\_email': 'zimmermannicole@williams.com',

'order\_price': '500.50',

'billing\_city':'Valier',

'product\_category':'health\_fitness',

'billing\_street':'69624 Erika Court',

'user\_agent':'Mozilla/5.0 (X11; Linux i686) AppleWebKit/532.1 (KHTML, like Gecko) Chrome/39.0.827.0 Safari/532.1',

'ip\_address':'183.83.29.194',

'card\_bin':'440955'

}

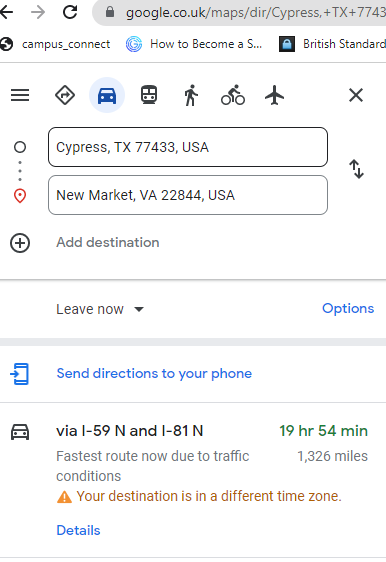
)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 640.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'ebe35d7d-a48e-483d-80ed-0c872f5a0c5b', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 19:55:04 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': 'ebe35d7d-a48e-483d-80ed-0c872f5a0c5b'}, 'RetryAttempts': 0}}

# **Testing Scenario- Transactions in another, distant place**



**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32124089d910036b127c7d3321a2c39d95',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '024-08-8261'

},

],

eventTimestamp='2022-01-08T03:53:34Z',

eventVariables={

'customer\_email': 'melissa34@nichols.com',

'order\_price': '18.55',

'billing\_city':'Cypress',

'product\_category':'health\_fitness',

'billing\_street':'258 Jonathan Inlet Suite 414',

'user\_agent':'Mozilla/5.0 (Linux; Android 4.2.1) AppleWebKit/536.2 (KHTML, like Gecko) Chrome/50.0.847.0 Safari/536.2',

'ip\_address':'167.63.61.48',

'card\_bin':'495601',

'billing\_zip':'77433'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 626.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '3451c818-b1bd-4b9b-a575-51153d626497', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Wed, 17 Aug 2022 07:38:43 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': '3451c818-b1bd-4b9b-a575-51153d626497'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='35624089d934036b127c7d3321a2c39d95',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '004-08-3461'

},

],

eventTimestamp='2022-01-12T03:53:34Z',

eventVariables={

'customer\_email': 'sarin@hotmail.com',

'order\_price': '89.55',

'billing\_city':'New Market',

'product\_category':'personal\_care',

'billing\_street':'121 Joseph Forge',

'user\_agent':'Mozilla/5.0 (X11; Linux i686) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/46.0.847.0 Safari/535.2',

'ip\_address':'71.92.221.49',

'card\_bin':'495601',

'billing\_zip':'22844'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 608.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '7f2ec641-6e95-4c36-a6db-0e85c14f116d', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Wed, 17 Aug 2022 07:43:46 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': '7f2ec641-6e95-4c36-a6db-0e85c14f116d'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='35624089d931246b127c7d3321a2c39d95',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '004-12-3461'

},

],

eventTimestamp='2022-01-12T03:53:34Z',

eventVariables={

'customer\_email': 'sarin123@hotmail.com',

'order\_price': '89.55',

'billing\_city':'New Market',

'product\_category':'grocery\_pos',

'billing\_street':'121 Joseph Forge',

'user\_agent':'Mozilla/5.0 (X11; Linux i686) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/46.0.847.0 Safari/535.2',

'ip\_address':'71.92.210.40',

'card\_bin':'495601',

'billing\_zip':'22844'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 421.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '9e9daf20-4d30-4b6e-bf22-5ae7b48fe1c1', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Wed, 17 Aug 2022 08:10:35 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': '9e9daf20-4d30-4b6e-bf22-5ae7b48fe1c1'}, 'RetryAttempts': 0}}

# **Testing scenario - large transactions**

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32bee5e620df4842340ef7ed1a6443a949',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '324-76-0162'

},

],

eventTimestamp='2022-01-14T11:30:20Z',

eventVariables={

'customer\_email': 'schmidtmolly@hotmail.com',

'order\_price': '56.63',

'billing\_city':'Lukeville',

'product\_category':'kids\_pets',

'billing\_street':'743 Jones Lodge',

'user\_agent':'Mozilla/5.0 (compatible; MSIE 5.0; Windows NT 4.0; Trident/4.0)',

'ip\_address':'21.163.68.239',

'card\_bin':'456841'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 616.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'cff93339-e02e-41e7-955e-290e945147dc', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 20:27:56 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': 'cff93339-e02e-41e7-955e-290e945147dc'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32bee5e62sd3f4842340ef7ed1a6443a949',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '324-76-0162'

},

],

eventTimestamp='2022-01-14T11:37:20Z',

eventVariables={

'customer\_email': 'schmidtmolly@hotmail.com',

'order\_price': '256.63',

'billing\_city':'Lukeville',

'product\_category':'kids\_pets',

'billing\_street':'743 Jones Lodge',

'user\_agent':'Mozilla/5.0 (compatible; MSIE 5.0; Windows NT 4.0; Trident/4.0)',

'ip\_address':'21.163.68.239',

'card\_bin':'456841'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 932.0}}], 'ruleResults': [{'ruleId': 'friction\_rule', 'outcomes': ['investigate']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '0c761e73-2cf9-4d3f-8039-4d605e5ef6e0', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 20:33:06 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '280', 'connection': 'keep-alive', 'x-amzn-requestid': '0c761e73-2cf9-4d3f-8039-4d605e5ef6e0'}, 'RetryAttempts': 0}}

# **Testing Scenario- Multiple transactions in a short period of time**

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32fc2341de67ac32dc6a13dd38bd3421ce1',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '596-83-8672'

},

],

eventTimestamp='2022-04-14T09:37:20Z',

eventVariables={

'customer\_email': 'ryancooper@brewer-campos.net',

'order\_price': '76.27',

'billing\_city':'Bloomington',

'product\_category':'health\_fitness',

'billing\_street':'15996 Jones Mountain',

'user\_agent':'Mozilla/5.0 (Linux; Android 3.2.3) AppleWebKit/535.0 (KHTML, like Gecko) Chrome/53.0.834.0 Safari/535.0',

'ip\_address':'32.250.190.177',

'card\_bin':'498045'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 181.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'fc857c10-7133-4907-96b4-8ab547dda4e6', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 21:31:54 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': 'fc857c10-7133-4907-96b4-8ab547dda4e6'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32fc2124de67ac32dc6a13dd38bd3421ce1',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '596-83-8672'

},

],

eventTimestamp='2022-04-14T09:42:20Z',

eventVariables={

'customer\_email': 'ryancooper@brewer-campos.net',

'order\_price': '196.27',

'billing\_city':'Bloomington',

'product\_category':'health\_fitness',

'billing\_street':'15996 Jones Mountain',

'user\_agent':'Mozilla/5.0 (Linux; Android 3.2.3) AppleWebKit/535.0 (KHTML, like Gecko) Chrome/53.0.834.0 Safari/535.0',

'ip\_address':'32.250.190.177',

'card\_bin':'498045'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 259.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': '7025a1fd-90dd-4260-86a3-206a31ea8b51', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 21:37:33 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': '7025a1fd-90dd-4260-86a3-206a31ea8b51'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32fc2124de13ac32dc6a13dd38bd3421ce1',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '596-83-8672'

},

],

eventTimestamp='2022-04-14T09:46:20Z',

eventVariables={

'customer\_email': 'ryancooper@brewer-campos.net',

'order\_price': '290.27',

'billing\_city':'Bloomington',

'product\_category':'health\_fitness',

'billing\_street':'15996 Jones Mountain',

'user\_agent':'Mozilla/5.0 (Linux; Android 3.2.3) AppleWebKit/535.0 (KHTML, like Gecko) Chrome/53.0.834.0 Safari/535.0',

'ip\_address':'32.250.190.177',

'card\_bin':'498045'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 664.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'eea5552c-c1f8-4644-894a-88d1b0138bec', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 21:38:44 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': 'eea5552c-c1f8-4644-894a-88d1b0138bec'}, 'RetryAttempts': 0}}

**Request:**

import boto3

fraudDetector = boto3.client('frauddetector')

response = fraudDetector.get\_event\_prediction(

detectorId='transaction\_fraud\_detector',

detectorVersionId='1',

eventId='32fc2198de13ac09dc6a13dd99bd3421ce1',

eventTypeName='transaction\_event',

entities=[

{

'entityType': 'customer',

'entityId': '596-83-8672'

},

],

eventTimestamp='2022-04-14T09:52:20Z',

eventVariables={

'customer\_email': 'ryancooper@brewer-campos.net',

'order\_price': '800.27',

'billing\_city':'Bloomington',

'product\_category':'health\_fitness',

'billing\_street':'15996 Jones Mountain',

'user\_agent':'Mozilla/5.0 (Linux; Android 3.2.3) AppleWebKit/535.0 (KHTML, like Gecko) Chrome/53.0.834.0 Safari/535.0',

'ip\_address':'32.250.190.177',

'card\_bin':'498045'

}

)

print(response)

**Response:**

{'modelScores': [{'modelVersion': {'modelId': 'transaction\_model', 'modelType': 'TRANSACTION\_FRAUD\_INSIGHTS', 'modelVersionNumber': '1.0'}, 'scores': {'transaction\_model\_insightscore': 735.0}}], 'ruleResults': [{'ruleId': 'approve\_rule', 'outcomes': ['approve']}], 'externalModelOutputs': [], 'ResponseMetadata': {'RequestId': 'abbc625c-8094-448e-8376-eaa19367031f', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 16 Aug 2022 21:40:59 GMT', 'content-type': 'application/x-amz-json-1.1', 'content-length': '275', 'connection': 'keep-alive', 'x-amzn-requestid': 'abbc625c-8094-448e-8376-eaa19367031f'}, 'RetryAttempts': 0}}