Predicting Loan Default Risk Using Census Income Data

Cloud-Deployed Income Classification API

Jonathan Chacko

Executive Summary

This project presents an end-to-end solution for **classifying individual loan applicants as low- or high-risk** using their demographic and socioeconomic information, leveraging the UCI Census Income dataset. A neural network classifier was developed and validated using robust cross-validation, and then deployed as a secure, production-ready API on Google Cloud Run using Docker. The report details the ML pipeline from preprocessing and model selection to cloud deployment, with screenshots verifying a fully operational cloud endpoint. The solution demonstrates the potential for financial institutions to modernize and automate credit risk assessment beyond traditional credit histories.

1. Introduction

Project Title: Predicting Loan Default Risk Using Census Income Data

Objective

To build a neural network model that classifies individuals as low-risk or high-risk loan applicants, based on demographic and socio-economic data from the UCI Census Income dataset.

Background

Financial institutions face the constant challenge of determining whether a loan applicant is likely to default. Traditionally, this involves credit history and financial statements—but what if we could leverage broader demographic and employment features to predict credit risk?

By classifying whether a person earns more than $50K per year using neural networks, we indirectly estimate their income capacity, a crucial factor in loan repayment ability. High-income individuals are typically lower-risk, while lower-income applicants may pose greater risk, especially without collateral or prior credit history.

2. Dataset Overview & Significance

Dataset Details

- **Primary Source:** [UCI Census Income Dataset]
- **Alternative Source:** [Kaggle Adult Census Income]
- **Shape:** ~48,842 rows, 14 features + target
- **Target Variable:** income (binary: >$50K or ≤$50K)

Note: Due to reliability issues with the UCI repository, the Kaggle copy was used for model development and deployment. Minor column differences were addressed during preprocessing.

Why This Project Matters

This work addresses a critical need in the financial sector: accurate, fair, and inclusive assessment of loan default risk. Unlike traditional methods that exclude applicants with little credit history, this model leverages broad demographic and socioeconomic features, expanding access to responsible credit.

3. Model Development Process

3.1 Data Preprocessing

- **Data Cleaning:**
  - Replaced '?' with NaN and dropped incomplete rows.
  - Standardized/cleaned string values.
  - Harmonized column names (especially for Kaggle import).
- **Feature Engineering:**
  - Age binned into: Young, Middle-aged, Old.
  - Created capital-profit (derived from capital-gain and capital-loss).
  - Dropped irrelevant columns (fnlwgt, education).
- **Target Encoding:**
  - Converted income to binary: 0 (≤$50K), 1 (> $50K).
- **Train/Test Split:**
  - 80% train/validation, 20% test; stratified for class balance.
- **Class Imbalance:**
  - Analyzed and addressed using class_weight in Keras (important since ~75% are ≤$50K).

3.2 Feature Preprocessing Pipeline

- **Numeric Pipeline:** Imputation (median) + StandardScaler
- **Categorical Pipeline:** Imputation ('missing') + OneHotEncoder
- **Combined with:** ColumnTransformer (from sklearn)

3.3 Model Training

- **Model:** Sequential Keras neural network with dropout, batch normalization, and LeakyReLU.
- **Cross-Validation:**
    - 10-fold K-Fold CV for robust validation
    - EarlyStopping on validation AUC
- **Final Model:**
    - Chosen as best fold (highest AUC); reloaded and saved as model.h5
    - Preprocessing pipeline exported as preprocessing_pipeline.pkl

3.4 Model Evaluation

Created Model (Detailed Evaluation):

- Excellent AUC: 0.90 indicates strong separation between income classes.
- Reasonable Accuracy: 0.78 indicates solid overall correctness.
- Moderate Precision: 0.53 suggests some false positives are expected.
- Excellent Recall: 0.90 shows the model captures nearly all actual positives.
- Precision vs Recall Trade-off: Precision = 0.53, Recall = 0.90
    - ➔ The model prioritizes catching more positives, even if some are incorrect. This is useful in high-income prediction scenarios where false negatives are costly.
- Consistent Cross-Validation: The model maintained highly stable performance across all 10 folds.

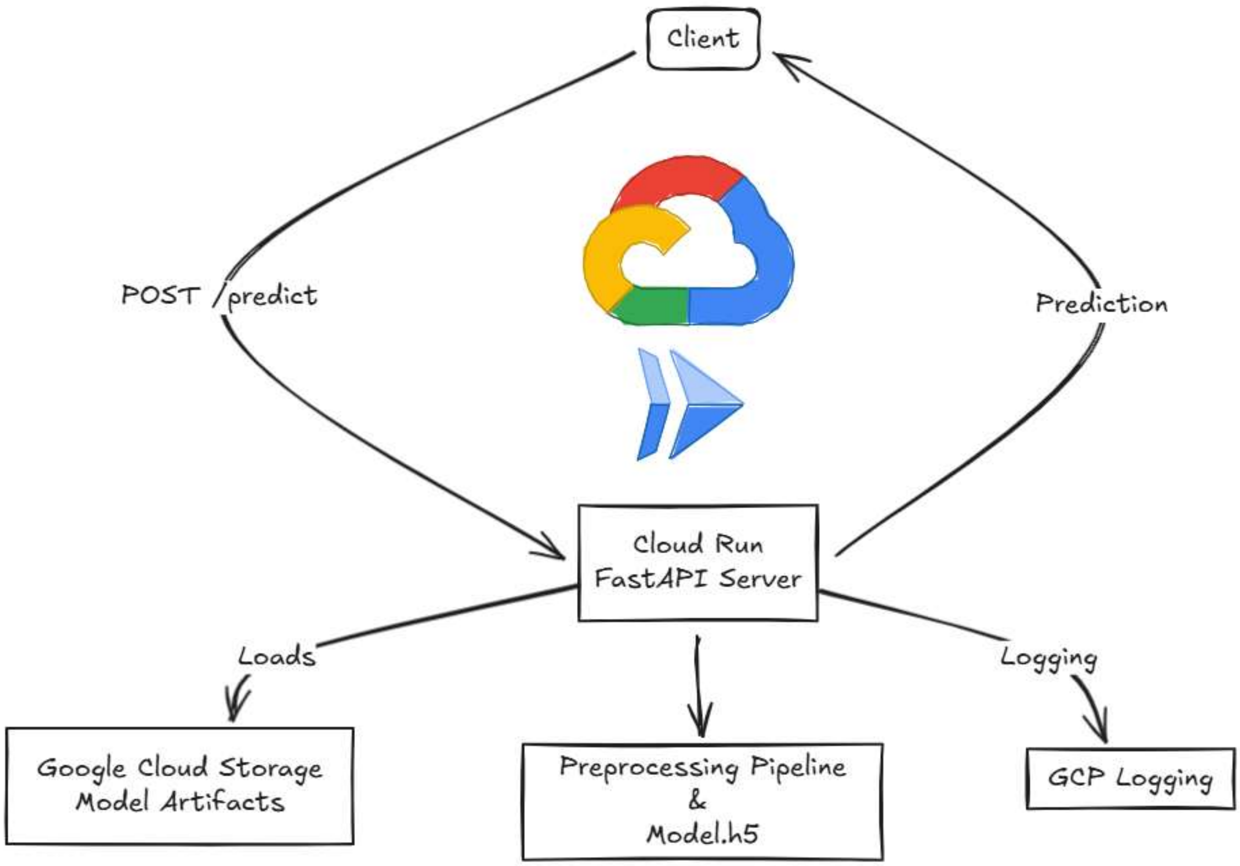
Deployed Model (with Test Set):

- Accuracy on Test Set: 77.87%

4. Deployment Architecture & Cloud Services

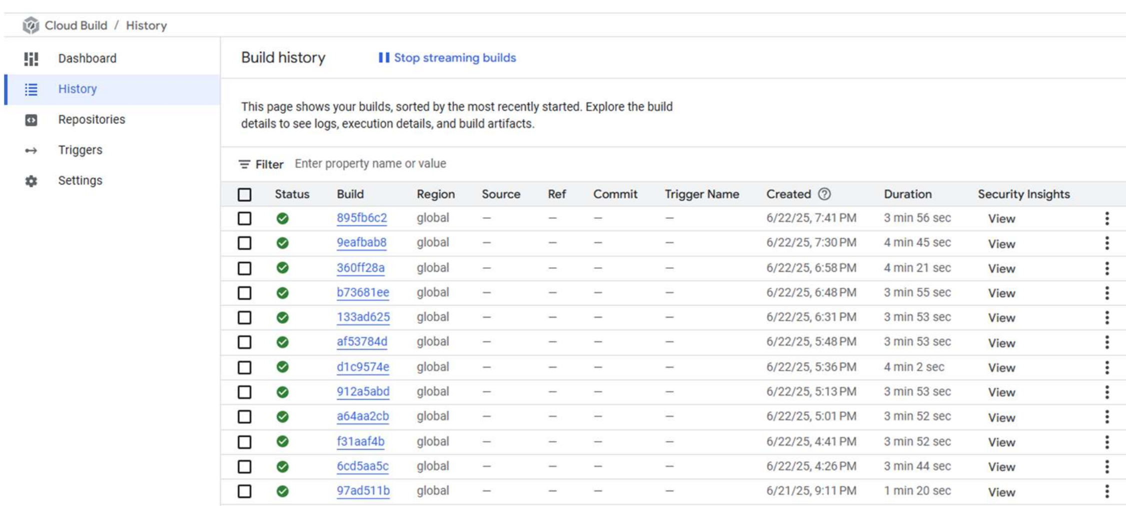
4.1 System Diagram

Deployment Architecture:



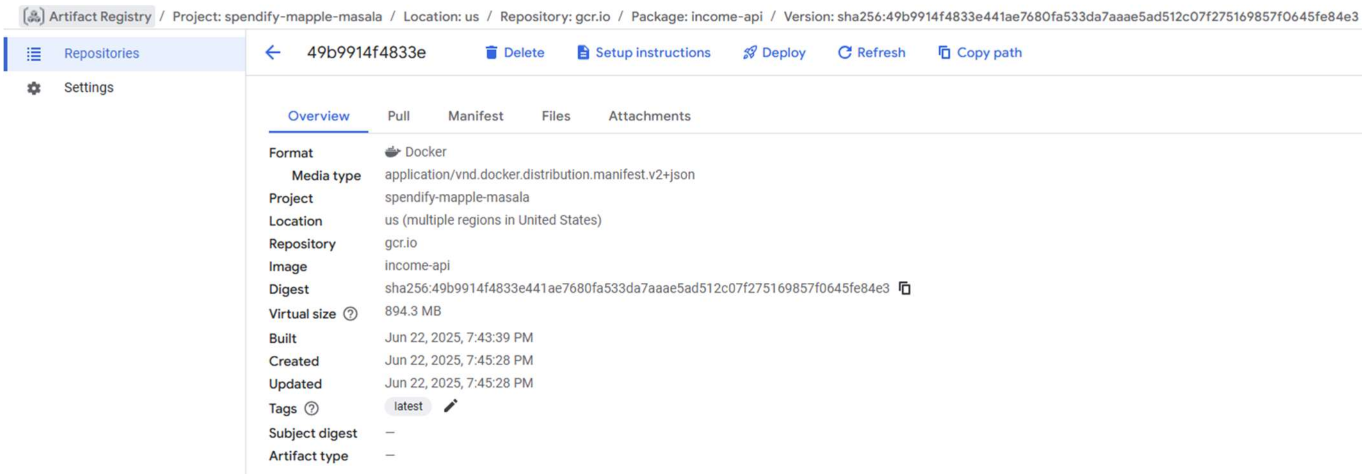
4.2 Cloud Services Used

- **Google Cloud Build:** For Docker image builds and registry push.

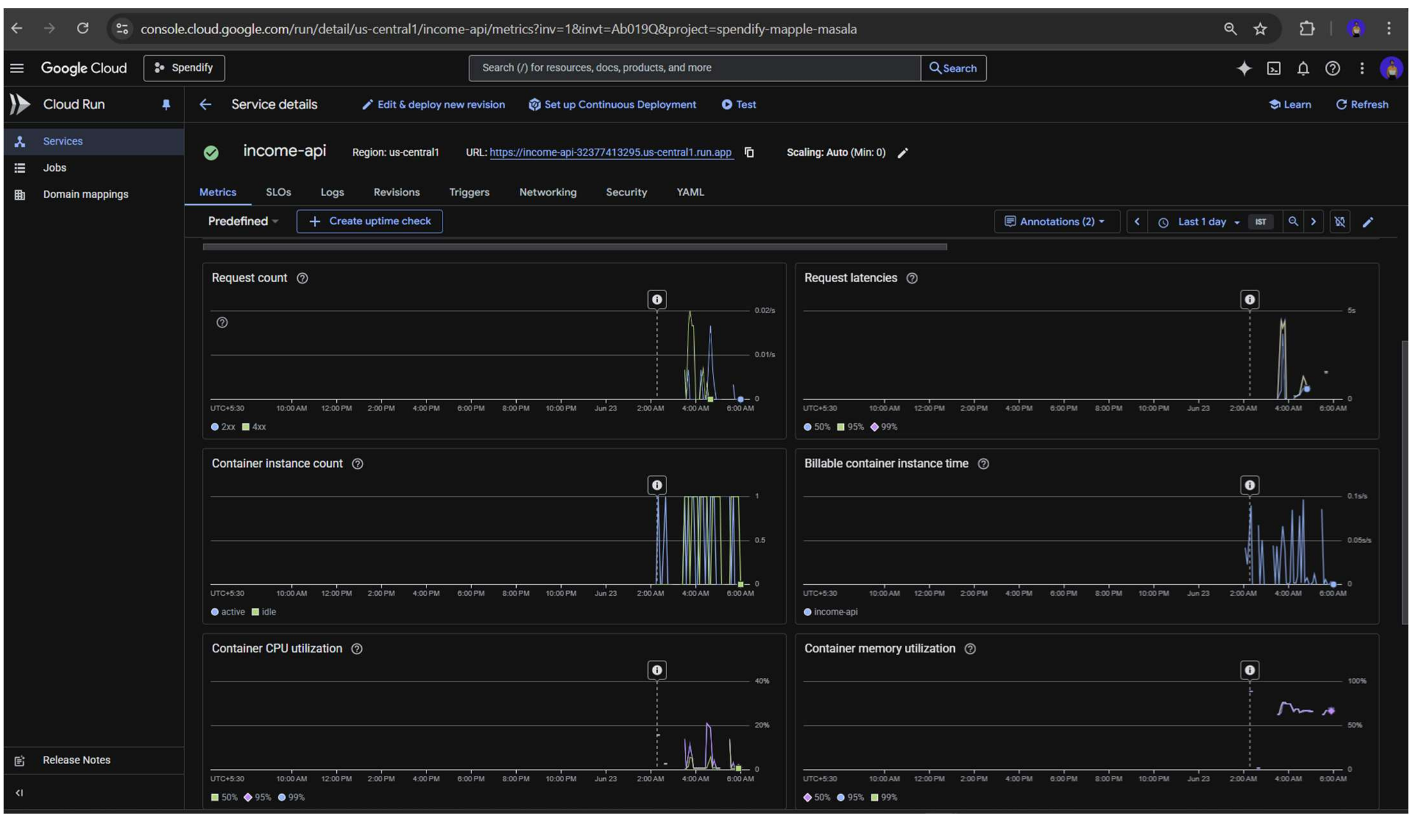


Status	Build	Region	Source	Ref	Commit	Trigger Name	Created	Duration	Security Insights
	895fb6c2	global	–	–	–	–	6/22/25, 7:41 PM	3 min 56 sec	View
	9eafbab8	global	–	–	–	–	6/22/25, 7:30 PM	4 min 45 sec	View
	360ff28a	global	–	–	–	–	6/22/25, 6:58 PM	4 min 21 sec	View
	b73681ee	global	–	–	–	–	6/22/25, 6:48 PM	3 min 55 sec	View
	133ad625	global	–	–	–	–	6/22/25, 6:31 PM	3 min 53 sec	View
	af53784d	global	–	–	–	–	6/22/25, 5:48 PM	3 min 53 sec	View
	d1c9574e	global	–	–	–	–	6/22/25, 5:36 PM	4 min 2 sec	View
	912a5abd	global	–	–	–	–	6/22/25, 5:13 PM	3 min 53 sec	View
	a64aa2cb	global	–	–	–	–	6/22/25, 5:01 PM	3 min 52 sec	View
	f31aaf4b	global	–	–	–	–	6/22/25, 4:41 PM	3 min 52 sec	View
	6cd5aa5c	global	–	–	–	–	6/22/25, 4:26 PM	3 min 44 sec	View
	97ad511b	global	–	–	–	–	6/21/25, 9:11 PM	1 min 20 sec	View

- **Google Container Registry:** Stores the container images.



- **Google Cloud Run:** Fully managed serverless deployment for the API.



4.3 Deployment & Verification

Build

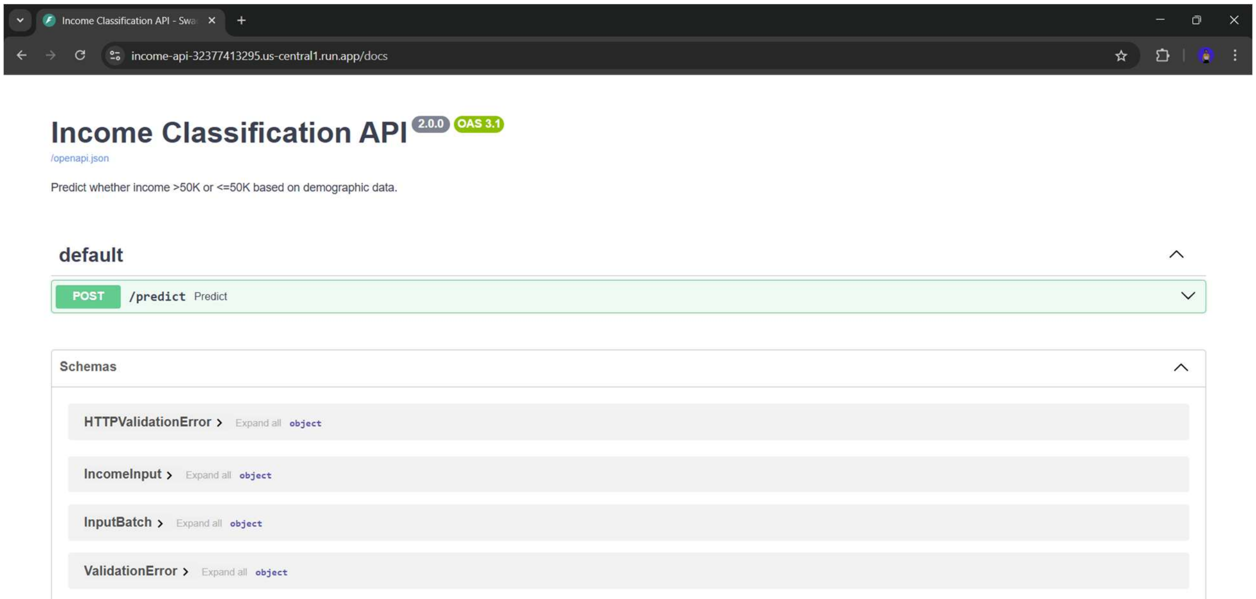
```
PS C:\Users\JonathanChackoPattas\OneDrive - Maritime Support Solutions\Desktop\Class Notes\Seneca\Semester 2\AIG200 - Capstone Project\Individual Submission
 - Machine Learning Model Deployment Assignment> gcloud builds submit --tag gcr.io/spendify-mapple-masala/income-api
Creating temporary archive of 13 file(s) totalling 6.1 MiB before compression.
Uploading tarball of [.] to [gs://spendify-mapple-masala_cloudbuild/source/1750624003.880277-2c6a85e22b4b4e76879782b8d12aaeda.tgz]
Created [https://cloudbuild.googleapis.com/v1/projects/spendify-mapple-masala/locations/global/builds/6cd5aa5c-1d3b-40d2-a1ca-82dc6b15327e].
Logs are available at [ https://console.cloud.google.com/cloud-build/builds/6cd5aa5c-1d3b-40d2-a1ca-82dc6b15327e?project=32377413295 ].
Waiting for build to complete. Polling interval: 1 second(s).
------------------------------------------------------------------ REMOTE BUILD OUTPUT ------------------------------------------------------------------
starting build "6cd5aa5c-1d3b-40d2-a1ca-82dc6b15327e"

FETCHSOURCE
Fetching storage object: gs://spendify-mapple-masala_cloudbuild/source/1750624003.880277-2c6a85e22b4b4e76879782b8d12aaeda.tgz#1750624006771733
Copying gs://spendify-mapple-masala_cloudbuild/source/1750624003.880277-2c6a85e22b4b4e76879782b8d12aaeda.tgz#1750624006771733...
/ [1 files][  4.2 MiB/  4.2 MiB]
Operation completed over 1 objects/4.2 MiB.
BUILD
Already have image (with digest): gcr.io/cloud-builders/docker
Sending build context to Docker daemon  6.445MB
Step 1/8 : FROM python:3.10-slim
```

```
Step 8/8 : CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8080"]
 ---> Running in 810be6602161
Removing intermediate container 810be6602161
 ---> 3bc9b325e7b3
Successfully built 3bc9b325e7b3
Successfully tagged gcr.io/spendify-mapple-masala/income-api:latest
PUSH
Pushing gcr.io/spendify-mapple-masala/income-api
The push refers to repository [gcr.io/spendify-mapple-masala/income-api]
35787cf71ea4: Preparing
8cb9c3c51ad8: Preparing
3ae046c80fb5: Preparing
3c4951c5548a: Preparing
63e09f79cfb7: Preparing
f3221a8c83dd: Preparing
e4e2acb8cf69: Preparing
7fb72a7d1a8e: Preparing
f3221a8c83dd: Waiting
e4e2acb8cf69: Waiting
7fb72a7d1a8e: Waiting
63e09f79cfb7: Layer already exists
f3221a8c83dd: Layer already exists
3c4951c5548a: Pushed
3ae046c80fb5: Pushed
e4e2acb8cf69: Layer already exists
7fb72a7d1a8e: Layer already exists
8cb9c3c51ad8: Pushed
35787cf71ea4: Pushed
latest: digest: sha256:f19399c6f451d7fe98279742dd343012b439f51f5dc5b8d000c9fe711c55386a size: 2002
DONE
--------------------------------------------------------------------------------------------------------------------------------------------------------
ID                                    CREATE_TIME                DURATION  SOURCE
                     IMAGES                                                        STATUS
6cd5aa5c-1d3b-40d2-a1ca-82dc6b15327e  2025-06-22T20:26:47+00:00  3M44S     gs://spendify-mapple-masala_cloudbuild/source/1750624003.880277-2c6a85e22b4b4e768
79782b8d12aaeda.tgz  gcr.io/spendify-mapple-masala/income-api (+1 more)  SUCCESS
PS C:\Users\JonathanChackoPattas\OneDrive - Maritime Support Solutions\Desktop\Class Notes\Seneca\Semester 2\AIG200 - Capstone Project\Individual Submission
 - Machine Learning Model Deployment Assignment> |
```

Deploy

```
PS C:\Users\JonathanChackoPattas\OneDrive - Maritime Support Solutions\Desktop\Class Notes\Seneca\Semester 2\AIG200 - Capstone Project\Individual Submission
 - Machine Learning Model Deployment Assignment> gcloud run deploy income-api --image gcr.io/spendify-mapple-masala/income-api --platform managed --region u
s-central1 --allow-unauthenticated --memory 1Gi
Deploying container to Cloud Run service [income-api] in project [spendify-mapple-masala] region [us-central1]
OK Deploying new service... Done.
  OK Creating Revision...
  OK Routing traffic...
  OK Setting IAM Policy...
Done.
Service [income-api] revision [income-api-00001-s8s] has been deployed and is serving 100 percent of traffic.
Service URL: https://income-api-32377413295.us-central1.run.app
PS C:\Users\JonathanChackoPattas\OneDrive - Maritime Support Solutions\Desktop\Class Notes\Seneca\Semester 2\AIG200 - Capstone Project\Individual Submission
 - Machine Learning Model Deployment Assignment> |
```

Test



- https://income-api-32377413295.us-central1.run.app/docs

Evaluate

- *API Endpoint:* https://income-api-32377413295.us-central1.run.app/predict
- *API Key for testing:* **idontknowit**

5. Challenges Faced

- **GCP Logging Difficulties:**
  Initially, logs were not visible in Cloud Run error pages. Using the command

  ```
  gcloud logging read "resource.type=cloud_run_revision AND
  resource.labels.service_name=income-api" --project=spendify-mapple-masala
  --limit=50 --freshness=1h --format="value(textPayload)"
  ```

  solved this for debugging build/runtime errors.

- **Dataset Sourcing:**
  The UCI link often failed; adaptation to Kaggle version was necessary, which included handling minor differences in column names and data format.

6. Conclusion & Future Work

This project demonstrates a robust, end-to-end machine learning workflow for real-world financial risk assessment using census income data. From rigorous model validation to secure, scalable cloud deployment, the solution highlights modern best practices for ML-driven APIs in the cloud.

Potential Improvements:

- Add automated re-training pipeline triggered by new data.
- Monitor model drift in production using GCP's AI Platform.
- Extend API to support batch predictions, streaming data, or richer applicant features.

7. References

1. Orginal Project: AIG100 – Project 3 => Capstone Individual Assignment
2. UCI Machine Learning Repository: Census Income Dataset (Currently Not Working)
3. Kaggle: Adult Census Income
4. **GitHub Repository:** *Project Code Repo* (**https://github.com/jcp-tech/Machine-Learning-Model-Deployment**)
    - **Live API:** https://income-api-32377413295.us-central1.run.app/predict
    - **Token for testing:** *idontknowit*