

# Loan Eligibility Prediction Project using Machine Learning on Google Cloud Platform

Dr. Aygul Zagidullina

March 14, 2024

# Introduction

Using Python to build a predictive model on GCP to determine whether an application requesting loan is eligible or not

What will we learn?

- What is loan eligibility prediction, and why is it important?
- What is GCP?
- What are the types of virtual machines available in GCP?
- How to create a virtual machine in GCP?
- How to set the IP address as Static for a VM instance in GCP?
- How to configure Jupyter in the GCP virtual machine?
- How to build Classifier models with Python?

# Problem Statement

## Business Objective:

The primary source of profit for the banking sector is from loan interests. Banks invest a lot of time and money in analyzing a loan applicant's profile. Going through the profile of every applicant manually is a highly tedious job. This is why automating this process is necessary.

**Aim:** To build a predictive model to identify if an applicant is capable of repaying the loan or not.

## Tech Stack:

Language: Python

Packages: Pandas, Numpy, Scikit-Learn, XGBoost, Imblearn

Cloud: GCP

# Instance Creation on GCP

3 ways:

Google Cloud Console

gcloud – command line tool (possibly next lecture)

Compute Engine API

← → ↺ console.cloud.google.com/compute/instancesAdd?project=dspro2-417019

Google Cloud DSPR02 Search (/)

### Create an instance

- New VM instance**  
Create a single VM instance from scratch  
[DEPLOY CONTAINER](#)
- New VM instance from template**  
Create a single VM instance from an existing template
- New VM instance from machine image**  
Create a single VM instance from an existing machine image
- Marketplace**  
Deploy a ready-to-go solution onto a VM instance

Deploy a container image to this VM instance

### Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

[PUBLIC IMAGES](#) [CUSTOM IMAGES](#) [SNAPSHOTS](#) [ARCHIVE SNAPSHOTS](#)

**Operating system**  
Ubuntu

**Version \***  
Ubuntu 20.04 LTS  
x86\_64, amd64 focal image built on 2024-03-07

**Boot disk type \***  
Balanced persistent disk

[COMPARE DISK TYPES](#)

**Size (GB) \***  
10  
Provision between 10 and 65536 GB

[SHOW ADVANCED CONFIGURATION](#)

[SELECT](#) [CANCEL](#)

### Identity and API access

**Service accounts**  
Service account  
Compute Engine default service account  
Requires the Service Account User role on the project.  
[Learn more](#)

**Access scopes**  
☒ Allow default access

# Instance Creation on GCP

Google Cloud

Technology areas ▾

Cross-product tools ▾

Related sites ▾

🔍 Search

Python

Overview

Guides

Reference

Samples

Filter

▸ The basics

▸ Get started with Google Cloud

▸ General AI, Generative AI, and ML guides

▸ General security guides

▸ Authentication

Accessibility

Keyboard shortcuts

Getting started

Setting up a Python development environment

Supported Python versions

Getting started with Python

Python 2 support on Google Cloud

▸ Quickstarts by use case

Official tutorials and solutions

App development

▸ Running app

▸ Storing data

▸ Logging and Debugging

▸ Developer tools

▸ Authentication

▸ Big data and machine learning

▸ Infrastructure and operations

Other supported languages

C#

Go

- Set up authentication (optional).

## Installing Python

Python's installation instructions vary by operating system. Follow the guide for the operating system you're running in your development environment, MacOS, Windows, or Linux.

macOS

Windows

Linux

Most Linux distributions include recent versions of Python.

1. To install Python in a Linux environment, install the appropriate packages for your distribution. For Debian and Ubuntu, these packages are [python3](#), and [python3-dev](#), and [python3-venv](#).

Install these packages using the following commands:

```
sudo apt update
sudo apt install python3 python3-dev python3-venv
```

2. You also need to install [pip](#). While Debian and most other distributions include a [python-pip](#) package, we recommend that you install [pip](#) to get the latest version:

```
sudo apt-get install wget
wget https://bootstrap.pypa.io/get-pip.py
sudo python3 get-pip.py
```

3. After the installations are complete, verify that you have [pip](#) installed:

```
pip3 --version
```

# Instance Creation on GCP: Installations

```
sudo apt update  
sudo apt install python3 python3-dev python3-venv  
  
sudo apt-get install wget  
wget https://bootstrap.pypa.io/get-pip.py  
sudo python3 get-pip.py  
  
pip3 --version
```

# Jupyter Notebook Setup

## How to enable Jupyter Notebook Viewing in GCP VM?

3 steps:

Make external IP a static one

Create Firewall rules

Jupyter notebook installations and configurations

← → ↺ console.cloud.google.com/net-security/firewall-manager/firewall-policies/add?project=dsp02-417019

Google Cloud DSP02 Search (/) for resources, docs, products, and more

**Network Security**

- Secure Web Proxy
- Cloud Armor
  - Cloud Armor policies
  - Adaptive Protection
  - Managed Protection
- Cloud IDS
  - IDS Dashboard
  - IDS Endpoints
  - IDS Threats
- Cloud NGFW
  - Firewall policies
  - Threats **PREVIEW**

**Create a firewall rule**

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Name \*

Lowercase letters, numbers, hyphens allowed

Description

**Logs**

Turning on firewall logs can generate a large number of logs which can increase costs in Logging. [Learn more](#)

☐ On

☒ Off

Network \*

default

Priority \*

1000 [CHECK PRIORITY OF OTHER FIREWALL RULES](#)

Priority can be 0 - 65535

# Jupyter Notebook Installations

```
sudo apt install python3-pip
```

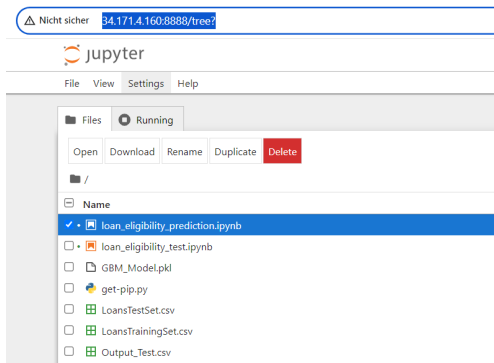
```
sudo pip3 install notebook
```

```
jupyter --version
```

```
jupyter-notebook --no-browser --port=8888
```

`http://External Static IP Address:Port Number`

`http://34.171.4.160:8888/`





# Jupyter Notebook Configurations


```
jupyter notebook --generate-config
```

```
vi /home/user name/.jupyter/jupyter_notebook_config.py
```

```
c.NotebookApp.ip = '*'
```

```
c.NotebookApp.open_browser = False
```

```
c.NotebookApp.port = 8888
```

 ssh.cloud.google.com/v2/ssh/projects/dspro2-417019/zones/us-central1-a/



SSH-in-browser

```
# Configuration file for notebook.

c = get_config() #noqa

c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
c.NotebookApp.port = 8888
#-----
# Application(SingletonConfigurable) configuration
#-----
## This is an application.
```

# Library Installations

pip install --upgrade wheel numpy pandas matplotlib seaborn  
pip install fancyimpute imblearn xgboost scikit-learn

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
import statistics
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.preprocessing import LabelBinarizer, StandardScaler, OrdinalEncoder
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from scipy.stats import boxcox
from sklearn.linear_model import LogisticRegression, RidgeClassifier, PassiveAggressiveClassifier
from sklearn import metrics
from sklearn import preprocessing
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from xgboost import plot_importance
from matplotlib import pyplot
from sklearn.naive_bayes import BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from fancyimpute import KNN, SoftImpute
from imblearn.over_sampling import SMOTE
import joblib
import operator
```