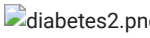
# Model Building and Analysis_Diabetes Dataset




## 1. Introduction :

- **In this Notbook, I will initially perform data preprocessing and analysis of the dataset. Secondly, we will split the dataset into train and test and will prepare it for the models to be designe. Next, I will design 5 major machine learning model and find the outputs of each model. Finally, I iwll do k-fold cross validation to check the consistency of the model and improve the results of the models as well.**

## 2. Importing libraries and loading the Dataset:

```
## Importing all necessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

%matplotlib inline

from collections import Counter
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

from sklearn.metrics import precision_score, accuracy_score, mean_absolute_error, mean_s

from imblearn.under_sampling import RandomUnderSampler, TomekLinks, AllKNN
from imblearn.under_sampling import InstanceHardnessThreshold, NeighbourhoodCleaningRule
from imblearn.over_sampling import RandomOverSampler, SMOTE , ADASYN, KMeansSMOTE, Borde
from imblearn.combine import SMOTETomek, SMOTEENN

import warnings
warnings.filterwarnings("ignore")


##Changing the resolution of figures :

plt.rcParams['figure.dpi']=80


## Loading the dataset
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))


     /kaggle/input/diabetes-prediction-dataset/diabetes_prediction_dataset.csv


# Reading the dataset
df = pd.read_csv('/kaggle/input/diabetes-prediction-dataset/diabetes_prediction_dataset.
df.head()
```

Release notes ×

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

### 2023-10-23

- Updated the **Open notebook** dialog for better usability and support for smaller screen sizes
- Added smart paste support for data from Google Sheets for R notebooks
- Enabled showing release notes in a tab
- Launched AI coding features for Pro/Pro+ users in Australia AU Canada CA India IN and Japan JP (tweet)
- Python package upgrades
  - earthengine-api 0.1.357 -> 0.1.375
  - flax 0.7.2 -> 0.7.4
  - geemap 0.27.4 -> 0.28.2
  - jax 0.4.14 -> 0.4.16
  - jaxlib 0.4.14 -> 0.4.16
  - keras 2.13.1 -> 2.14.0
  - tensorboard 2.13.0 -> 2.14.1
  - tensorflow 2.13.0 -> 2.14.0
  - tensorflow-gcs-config 2.13.0 -> 2.14.0
  - tensorflow-hub 0.14.0 -> 0.15.0
  - tensorflow-probability 0.20.1 -> 0.22.0
  - torch 2.0.1 -> 2.1.0
  - torchaudio 2.0.2 -> 2.1.0
  - torchtext 0.15.2 -> 0.16.0
  - torchvision 0.15.2 -> 0.16.0
  - xgboost 1.7.6 -> 2.0.0
- Python package inclusions
  - bigframes 0.10.0
  - malloy 2023.1056

### 2023-09-22

- Added the ability to scope an AI generated suggestion to a specific Pandas dataframe (tweet)
- Added Colab link previews to Docs (tweet)
- Added smart paste support for data from Google Sheets
- Increased font size of dropdowns in interactive forms
- Improved rendering of the notebook when printing
- Python package upgrades
  - tensorflow 2.12.0 -> 2.13.0
  - tensorboard 2.12.3 -> 2.13.0
  - keras 2.12.0 -> 2.13.1
  - tensorflow-gcs-config 2.12.0 -> 2.13.
  - scipy 1.10.1-> 1.11.2
  - cython 0.29.6 -> 3.0.2
- Python package inclusions
  - geemap 0.26.0

### 2023-08-18

- Added "Change runtime type" to the menu in the connection button
- Improved auto-reconnection to an already running notebook (#3764)
- Increased the specs of our highmem machines for Pro users
- Fixed `add-apt-repository` command on Ubuntu 22.04 runtime (#3867)
- Python package upgrades
  - bokeh 2.4.3 -> 3.2.2
  - cmake 3.25.2 -> 3.27.2
  - cryptography 3.4.8 -> 41.0.3
  - dask 2022.12.1 -> 2023.8.0
  - distributed 2022.12.1 -> 2023.8.0
  - earthengine-api 0.1.358 -> 0.1.364
  - flax 0.7.0 -> 0.7.2
  - ipython-sql 0.4.0 -> 0.5.0
  - jax 0.4.13 -> 0.4.14
  - jaxlib 0.4.13 -> 0.4.14
  - lightgbm 3.3.5 -> 4.0.0
  - mkl 2019.0 -> 2023.2.0
  - notebook 6.4.8 -> 6.5.5
  - numpy 1.22.4 -> 1.23.5
  - opencv-python 4.7.0.72 -> 4.8.0.76
  - pillow 8.4.0 -> 9.4.0

| | gender | age | hypertension | heart_disease | smoking_history | bmi | HbA1c_level | b |
|---|---|---|---|---|---|---|---|---|
| 0 | Female | 80.0 | 0 | 1 | never | 25.19 | 6.6 | |
| 1 | Female | 54.0 | 0 | 0 | No Info | 27.32 | 6.6 | |
| 2 | Male | 28.0 | 0 | 0 | never | 27.32 | 5.7 | |
| 3 | Female | 36.0 | 0 | 0 | current | 23.45 | 5.0 | |
| 4 | Male | 76.0 | 1 | 1 | current | 20.14 | 4.8 | |

## 3. Cleaning and Preprocessing the Dataset:

```
##data size

df.shape
```

```
(100000, 9)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   gender               100000 non-null  object 
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64  
 3   heart_disease        100000 non-null  int64  
 4   smoking_history      100000 non-null  object 
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64  
 8   diabetes             100000 non-null  int64  
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

- No null values are present in this dataset.

```
df.describe()
```

| | age | hypertension | heart_disease | bmi | HbA1c_level | bl |
|---|---|---|---|---|---|---|
| count | 100000.000000 | 100000.00000 | 100000.000000 | 100000.000000 | 100000.000000 | |
| mean | 41.885856 | 0.07485 | 0.039420 | 27.320767 | 5.527507 | |
| std | 22.516840 | 0.26315 | 0.194593 | 6.636783 | 1.070672 | |
| min | 0.080000 | 0.00000 | 0.000000 | 10.010000 | 3.500000 | |
| 25% | 24.000000 | 0.00000 | 0.000000 | 23.630000 | 4.800000 | |
| 50% | 43.000000 | 0.00000 | 0.000000 | 27.320000 | 5.800000 | |
| 75% | 60.000000 | 0.00000 | 0.000000 | 29.580000 | 6.200000 | |
| max | 80.000000 | 1.00000 | 1.000000 | 95.690000 | 9.000000 | |

```
# duplicates records :-
df.duplicated().sum()
```

```
3854
```

Although, there are 3854 duplicate rows from the 100,000 rows, we can't ignore/remove these from the dataset as there may be different peoples who has same reding of different features such as age, hypertension, bmi etc

## 4. Exploration of Data:

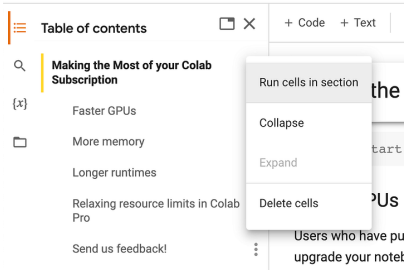
- plotly 5.13.1 -> 5.15.0
- prettytable 0.7.2 -> 3.8.0
- pytensor 2.10.1 -> 2.14.2
- spacy 3.5.4 -> 3.6.1
- statsmodels 0.13.5 -> 0.14.0
- xarray 2022.12.0 -> 2023.7.0

- Python package inclusions
  - PyDrive2 1.6.3

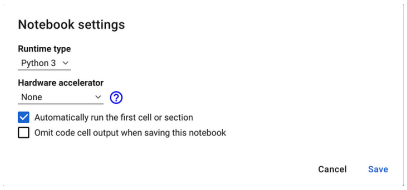
### 2023-07-21

- Launched auto-plotting for dataframes, available using the chart button that shows up alongside datatables (post)

- Added a menu to the table of contents to support running a section or collapsing/expanding sections (post)




- Added an option to automatically run the first cell or section, available under Edit -> Notebook settings (post)




- Launched Pro/Pro+ to Algeria, Argentina, Chile, Ecuador, Egypt, Ghana, Kenya, Malaysia, Nepal, Nigeria, Peru, Rwanda, Saudi Arabia, South Africa, Sri Lanka, Tunisia, and Ukraine (tweet)
- Added a command, "Toggle tab moves focus" for toggling tab trapping in the editor (Tools -> Command palette, "Toggle tab moves focus")
- Fixed issue where `files.upload()` was sometimes returning an incorrect filename (#1550)
- Fixed f-string syntax highlighting bug (#3802)
- Disabled ambiguous characters highlighting for commonly used LaTeX characters (#3648)
- Upgraded Ubuntu from 20.04 LTS to 22.04 LTS
- Updated the Colab Marketplace VM image
- Python package upgrades:
  - autograd 1.6.1 -> 1.6.2
  - drivefs 76.0 -> 77.0
  - flax 0.6.11 -> 0.7.0
  - earthengine-api 0.1.357 -> 0.1.358
  - GDAL 3.3.2->3.4.3
  - google-cloud-bigquery-storage 2.20.0 -> 2.22.2
  - gspread-dataframe 3.0.8 -> 3.3.1
  - holidays 0.27.1 -> 0.29
  - jax 0.4.10 -> jax 0.4.13
  - jaxlib 0.4.10 -> jax 0.4.13
  - jupyterlab-widgets 3.0.7 -> 3.0.8
  - nbformat 5.9.0 -> 5.9.1
  - opencv-python-headless 4.7.0.72 -> 4.8.0.74
  - pygame 2.4.0 -> 2.5.0
  - spacy 3.5.3 -> 3.5.4
  - SQLAlchemy 2.0.16 -> 2.0.19
  - tabulate 0.8.10 -> 0.9.0
  - tensorflow-hub 0.13.0 -> 0.14.0

### 2023-06-23

- Launched AI coding features to subscribed users starting with Pro+ users in the US (tweet, post)

## a. Univariate Analysis :

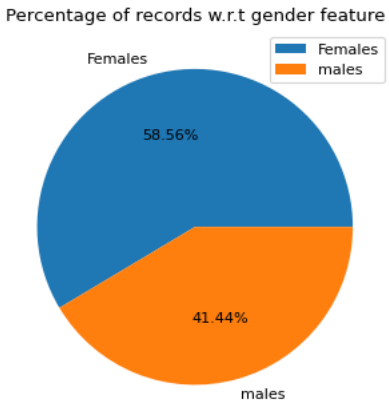
```
# gender :
df['gender'].value_counts()
```

```
gender
Female    58552
Male      41430
Other        18
Name: count, dtype: int64
```

```
# removing records with gender not defined "low number of records which will not affect
df = df[df['gender']!='Other']

num_var = list(df['gender'].value_counts().values)
labels = list(df['gender'].value_counts().index)
count_dict = dict(enumerate(num_var))

print(num_var)
print(labels)
```

```
[58552, 41430]
['Female', 'Male']
```

```
# Gender visualization
plt.pie(count_dict.values(), labels = ['Females', 'males'],autopct='%1.2f%%');
plt.legend();
plt.title('Percentage of records w.r.t gender feature');
```




```
# age :
plt.figure(figsize = [12,6]);
plt.hist(df['age'], bins = 40);
plt.xlabel('Ages');
plt.ylabel('Counts');
plt.xticks(np.arange(0,82,2));
```

- Added the Kernel Selector in the Notebook Settings (tweet)
- Fixed double space trimming issue in markdown #3766
- Fixed run button indicator not always centered #3609
- Fixed inconsistencies for automatic indentation on multi-line #3697
- Upgraded Python from 3.10.11 to 3.10.12
- Python package updates:
  - duckdb 0.7.1 -> 0.8.1
  - earthengine-api 0.1.350 -> 0.1.357
  - flax 0.6.9 -> 0.6.11
  - google-cloud-bigquery 3.9.0 -> 3.10.0
  - google-cloud-bigquery-storage 2.19.1 -> 2.20.0
  - grpcio 1.54.0 -> 1.56.0
  - holidays 0.25 -> 0.27.1
  - nbformat 5.8.0 -> 5.9.0
  - prophet 1.1.3 -> 1.1.4
  - pydata-google-auth 1.7.0 -> 1.8.0
  - spacy 3.5.2 -> 3.5.3
  - tensorboard 2.12.2 -> 2.12.3
  - xgboost 1.7.5 -> 1.7.6
- Python package inclusions:
  - gcsfs 2023.6.0
  - geopandas 0.13.2
  - google-cloud-bigquery-connection 1.12.0
  - google-cloud-functions 1.13.0
  - grpc-google-iam-v1 0.12.6
  - multidict 6.0.4
  - tensorboard-data-server 0.7.1

### 2023-06-02

- Released the new site colab.google
- Published Colab's Docker runtime image to us-docker.pkg.dev/colab-images/public/runtime (tweet, instructions)
- Launched support for Google children accounts (tweet)
- Launched DagsHub integration (tweet, post)
- Upgraded to Monaco Editor Version 0.37.1
- Fixed various Vim keybinding bugs
- Fixed issue where the N and P letters sometimes couldn't be typed (#3664)
- Fixed rendering support for compositional inputs (#3660, #3679)
- Fixed lag in notebooks with lots of cells (#3676)
- Improved support for R by adding a Runtime type notebook setting (Edit -> Notebook settings)
- Improved documentation for connecting to a local runtime (Connect -> Connect to a local runtime)
- Python package updates:
  - holidays 0.23 -> 0.25
  - jax 0.4.8 -> 0.4.10
  - jaxlib 0.4.8 -> 0.4.10
  - pip 23.0.1 -> 23.1.2
  - tensorflow-probability 0.19.0 -> 0.20.1
  - torch 2.0.0 -> 2.0.1
  - torchaudio 2.0.1 -> 2.0.2
  - torchdata 0.6.0 -> 0.6.1
  - torchtext 0.15.1 -> 0.15.2
  - torchvision 0.15.1 -> 0.15.2
  - tornado 6.2 -> 6.3.1

### 2023-05-05

- Released GPU type selection for paid users, allowing them to choose a preferred NVidia GPU
- Upgraded R from 4.2.3 to 4.3.0
- Upgraded Python from 3.9.16 to 3.10.11
- Python package updates:
  - attrs 22.2.0 -> attrs 23.1.0
  - earthengine-api 0.1.349 -> earthengine-api 0.1.350
  - flax 0.6.8 -> 0.6.9
  - grpcio 1.53.0 -> 1.54.0
  - nbclient 0.7.3 -> 0.7.4
  - tensorflow-datasets 4.8.3 -> 4.9.2
  - termcolor 2.2.0 -> 2.3.0
  - zict 2.2.0 -> 3.0.0

### 2023-04-14

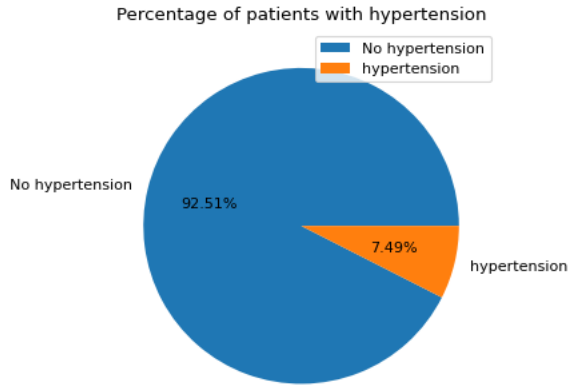
- Python package updates:
  - google-api-python-client 2.70.0 -> 2.84.0


```
# hypertension :
df['hypertension'].value_counts()
```

```
     hypertension
     0    92497
     1     7485
     Name: count, dtype: int64
```



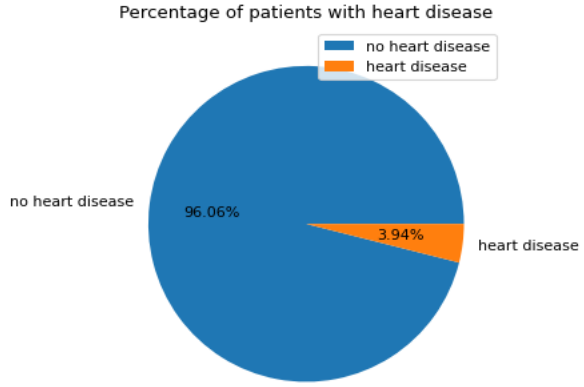
```
num_var = list(df['hypertension'].value_counts().values)
count_dict = dict(enumerate(num_var))

# [0] return to 'no hypertension' and [1] return to 'hypertension'
plt.pie(count_dict.values(), labels = ['No hypertension', 'hypertension'],autopct='%1.2f
plt.legend();
plt.title('Percentage of patients with hypertension');
```




```
# heart_disease :
df['heart_disease'].value_counts()
```

```
     heart_disease
     0    96040
     1     3942
     Name: count, dtype: int64
```

```
num_var = list(df['heart_disease'].value_counts().values)
count_dict = dict(enumerate(num_var))

# [0] return to 'no hypertension' and [1] return to 'hypertension'
plt.pie(count_dict.values(), labels = ['no heart disease', 'heart disease'],autopct='%1.
plt.legend();
plt.title('Percentage of patients with heart disease');
```




```
# smoking_history :
df['smoking_history'].value_counts()
```

- google-auth-oauthlib 0.4.6 -> 1.0.0
- google-cloud-bigquery 3.4.2 -> 3.9.0
- google-cloud-datastore 2.11.1 -> 2.15.1
- google-cloud-firestore 2.7.3 -> 2.11.0
- google-cloud-language 2.6.1 -> 2.9.1
- google-cloud-storage 2.7.0 -> 2.8.0
- google-cloud-translate 3.8.4 -> 3.11.1
- networkx 3.0 -> 3.1
- notebook 6.3.0 -> 6.4.8
- jax 0.4.7 -> 0.4.8
- pandas 1.4.4 -> 1.5.3
- spacy 3.5.1 -> 3.5.2
- SQLAlchemy 1.4.47 -> 2.0.9
- xgboost 1.7.4 -> 1.7.5

### 2023-03-31

- Improve bash ! syntax highlighting (GitHub issue)
- Fix bug where VIM keybindings weren't working in the file editor
- Upgraded R from 4.2.2 to 4.2.3
- Python package updates:
  - arviz 0.12.1 --> 0.15.1
  - astropy 4.3.1 --> 5.2.2
  - dopamine-rl 1.0.5 --> 4.0.6
  - gensim 3.6.0 --> 4.3.1
  - ipykernel 5.3.4 -> 5.5.6
  - ipython 7.9.0 -> 7.34.0
  - jax 0.4.4 -> 0.4.7
  - jaxlib 0.4.4 -> 0.4.7
  - jupyter_core 5.2.0 -> 5.3.0
  - keras 2.11.0 -> 2.12.0
  - lightgbm 2.2.3 -> 3.3.5
  - matplotlib 3.5.3 -> 3.7.1
  - nltk 3.7 -> 3.8.1
  - opencv-python 4.6.0.66 -> 4.7.0.72
  - plotly 5.5.0 -> 5.13.1
  - pymc 4.1.4 -> 5.1.2
  - seaborn 0.11.2 -> 0.12.2
  - spacy 3.4.4 -> 3.5.1
  - sympy 1.7.1 -> 1.11.1
  - tensorboard 2.11.2 -> 2.12.0
  - tensorflow 2.11.0 -> 2.12.0
  - tensorflow-estimator 2.11.0 -> 2.12.0
  - tensorflow-hub 0.12.0 -> 0.13.0
  - torch 1.13.1 -> 2.0.0
  - torchaudio 0.13.1 -> 2.0.1
  - torchtext 0.14.1 -> 0.15.1
  - torchvision 0.14.1 -> 0.15.1

### 2023-03-10

- Added the Colab editor shortcuts example notebook
- Fixed triggering of @-mention and email autocomplete for large comments (GitHub issue)
- Added View Resources to the Runtime menu
- Made file viewer images fit the view by default, resizing to original size on click
- When in VIM mode, enable copy as well as allowing propagation to monaco-vim to escape visual mode (GitHub issue)
- Upgraded CUDA 11.6.2 -> 11.8.0 and cuDNN 8.4.0.27 -> 8.7.0.84
- Upgraded Nvidia drivers 525.78.01 -> 530.30.02
- Upgraded Python 3.8.10 -> 3.9.16
- Python package updates:
  - beautifulsoup4 4.6.3 -> 4.9.3
  - bokeh 2.3.3 -> 2.4.3
  - debugpy 1.0.0 -> 1.6.6
  - Flask 1.1.4 -> 2.2.3
  - jax 0.3.25 -> 0.4.4
  - jaxlib 0.3.25 -> 0.4.4
  - Jinja2 2.11.3 -> 3.1.2
  - matplotlib 3.2.2 -> 3.5.3
  - nbconvert 5.6.1 -> 6.5.4
  - pandas 1.3.5 -> 1.4.4
  - pandas-datareader 0.9.0 -> 0.10.0
  - pandas-profiling 1.4.1 -> 3.2.0
  - Pillow 7.1.2 -> 8.4.0
  - plotnine 0.8.0 -> 0.10.1
  - scikit-image 0.18.3 -> 0.19.3
  - scikit-learn 1.0.2 -> 1.2.2
  - scipy 1.7.3 -> 1.10.1
  - setuptools 57.4.0 -> 63.4.3

```
smoking_history
No Info        35810
never          35092
former          9352
current         9286
not current     6439
ever            4003
Name: count, dtype: int64
```

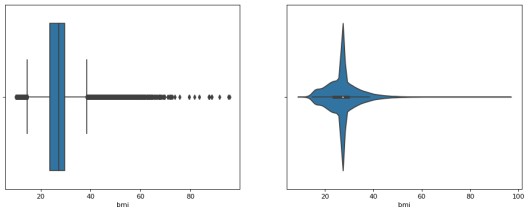
```
df[df['smoking_history'] == 'No Info'].shape[0] / df.shape[0] *100
```

```
35.81644696045288
```

- 35.81% of records have **no info** about smoking history, which is considerably high. Thus, we may remove this column to maintain better data integrity

```
# removing smoking_history column :
df = df.drop(columns = ['smoking_history'])
```

```
# bmi :
plt.figure(figsize = [14,5]);
plt.subplot(1,2,1);
sns.boxplot(data = df, x = 'bmi');

plt.subplot(1,2,2);
sns.violinplot(data = df, x = 'bmi');
```




```
# It is clear from the plot that, there are outliers which we need to remove.
# 1. frist calculation IQR
IQR = df['bmi'].quantile(0.75) - df['bmi'].quantile(0.25)

# determine the upper and lower boundries :
upper_boundry = df['bmi'].quantile(0.75) + (IQR*1.5)
lower_boundry = df['bmi'].quantile(0.25) - (IQR*1.5)

# removing outliers :
df = df[df['bmi'] <= upper_boundry]
df = df[df['bmi'] >= lower_boundry]

sns.violinplot(data = df, x = 'bmi');
```

- sklearn-pandas 1.8.0 -> 2.2.0
- statsmodels 0.12.2 -> 0.13.5
- urllib3 1.24.3 -> 1.26.14
- Werkzeug 1.0.1 -> 2.2.3
- wrapt 1.14.1 -> 1.15.0
- xgboost 0.90 -> 1.7.4
- xlrd 1.2.0 -> 2.0.1

### 2023-02-17

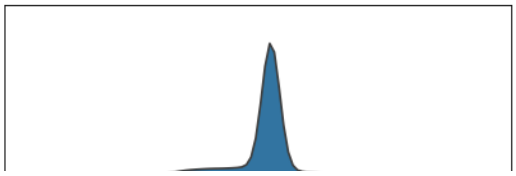
- Show graphs of RAM and disk usage in notebook toolbar
- Copy cell links directly to the clipboard instead of showing a dialog when clicking on the link icon in the cell toolbar
- Updated the Colab Marketplace VM image
- Upgraded CUDA to 11.6.2 and cuDNN to 8.4.0.27
- Python package updates:
  - tensorflow 2.9.2 -> 2.11.0
  - tensorboard 2.9.1 -> 2.11.2
  - keras 2.9.0 -> 2.11.0
  - tensorflow-estimator 2.9.0 -> 2.11.0
  - tensorflow-probability 0.17.0 -> 0.19.0
  - tensorflow-gcs-config 2.9.0 -> 2.11.0
  - earthengine-api 0.1.339 -> 0.1.341
  - flatbuffers 1.12 -> 23.1.21
  - platformdirs 2.6.2 -> 3.0.0
  - pydata-google-auth 1.6.0 -> 1.7.0
  - python-utils 3.4.5 -> 3.5.2
  - tenacity 8.1.0 -> 8.2.1
  - tifffile 2023.1.23.1 -> 2023.2.3
  - notebook 5.7.16 -> 6.3.0
  - tornado 6.0.4 -> 6.2
  - aiohttp 3.8.3 -> 3.8.4
  - charset-normalizer 2.1.1 -> 3.0.1
  - fastai 2.7.0 -> 2.7.1
  - soundfile 0.11.0 -> 0.12.1
  - typing-extensions 4.4.0 -> 4.5.0
  - widgetsnbextension 3.6.1 -> 3.6.2
  - pydantic 1.10.4 -> 1.10.5
  - zipp 3.12.0 -> 3.13.0
  - numpy 1.21.6 -> 1.22.4
  - drivefs 66.0 -> 69.0
  - gdal 3.0.4 -> 3.3.2 GitHub issue
- Added libudunits2-dev for smoother R package installs GitHub issue

### 2023-02-03

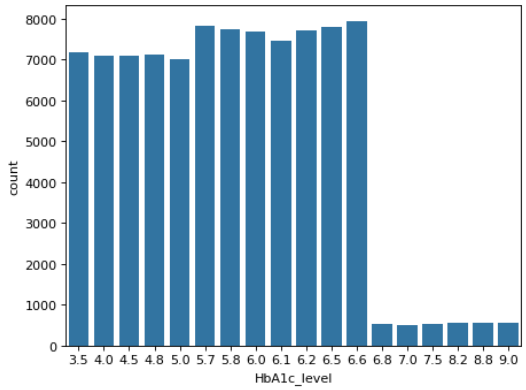
- Improved tooltips for pandas series to show common statistics about the series object
- Made the forms dropdown behave like an autocomplete box when it allows input
- Updated the nvidia driver from 460.32.03 to 510.47.03
- Python package updates:
  - absl-py 1.3.0 -> 1.4.0
  - bleach 5.0.1 -> 6.0.0
  - cachetools 5.2.1 -> 5.3.0
  - cmdstanpy 1.0.8 -> 1.1.0
  - dnspython 2.2.1 -> 2.3.0
  - fsspec 2022.11.0 -> 2023.1.0
  - google-cloud-bigquery-storage 2.17.0 -> 2.18.1
  - holidays 0.18 -> 0.19
  - jupyter-core 5.1.3 -> 5.2.0
  - packaging 21.3 -> 23.0
  - prometheus-client 0.15.0 -> 0.16.0
  - pyct 0.4.8 -> 0.5.0
  - pydata-google-auth 1.5.0 -> 1.6.0
  - python-slugify 7.0.0 -> 8.0.0
  - sqlalchemy 1.4.46 -> 2.0.0
  - tensorflow-io-gcs-filesystem 0.29.0 -> 0.30.0
  - tifffile 2022.10.10 -> 2023.1.23.1
  - zipp 3.11.0 -> 3.12.0
  - Pinned sqlalchemy to version 1.4.46

### 2023-01-12

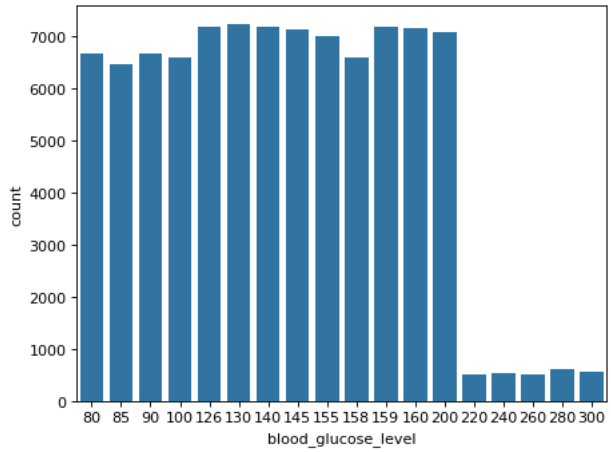
- Added support for @-mention and email autocomplete in comments
- Improved errors when GitHub notebooks can't be loaded
- Increased color contrast for colors used for syntax highlighting in the code editor
- Added terminal access for custom GCE VM runtimes



```
# HbA1c_level :
base_color = sns.color_palette()[0]
sns.countplot(data = df, x = df['HbA1c_level'], color = base_color);
```




```
# blood_glucose_level :
sns.countplot(data = df, x = df['blood_glucose_level'], color = base_color);
```




```
# diabetes :-
labels = list(df['diabetes'].value_counts().index)
num_var = list(df['diabetes'].value_counts().values)
print(labels)
print(num_var)
```

```
[0, 1]
[85875, 7022]
```

```
# [0] return to 'no diabetes' and [1] return to 'diabetes'
plt.pie(num_var, labels = ['no diabetes', 'diabetes'],autopct='%1.2f%%');
plt.legend();
plt.title('Visualizing percentage of patients have diabetes');
```

- Upgraded Ubuntu from 18.04 LTS to 20.04 LTS (GitHub issue)
- Python package updates:
  - GDAL 2.2.2 -> 2.2.3.
  - NumPy from 1.21.5 to 1.21.6.
  - attrs 22.1.0 -> 22.2.0
  - chardet 3.0.4 -> 4.0.0
  - cloudpickle 1.6.0 -> 2.2.0
  - filelock 3.8.2 -> 3.9.0
  - google-api-core 2.8.2 -> 2.11.0
  - google-api-python-client 1.12.11 -> 2.70.0
  - google-auth-httplib2 0.0.3 -> 0.1.0
  - google-cloud-bigquery 3.3.5 -> 3.4.1
  - google-cloud-datastore 2.9.0 -> 2.11.0
  - google-cloud-firestore 2.7.2 -> 2.7.3
  - google-cloud-storage 2.5.0 -> 2.7.0
  - holidays 0.17.2 -> holidays 0.18
  - importlib-metadata 5.2.0 -> 6.0.0
  - networkx 2.8.8 -> 3.0
  - opencv-python-headless 4.6.0.66 -> 4.7.0.68
  - pip 21.1.3 -> 22.04
  - pip-tools 6.2.0 -> 6.6.2
  - prettytable 3.5.0 -> 3.6.0
  - requests 2.23.0 -> 2.25.1
  - termcolor 2.1.1 -> 2.2.0
  - torch 1.13.0 -> 1.13.1
  - torchaudio 0.13.0 -> 0.13.1
  - torchtext 0.14.0-> 0.14.1
  - torchvision 0.14.0 -> 0.14.1

### 2022-12-06

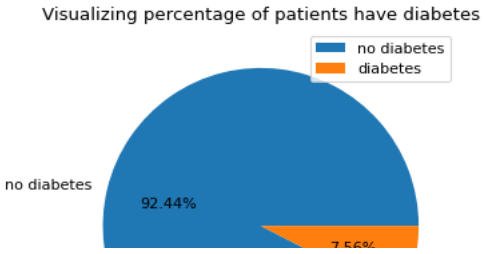
- Made fallback runtime version available until mid-December (GitHub issue)
- Upgraded to Python 3.8 (GitHub issue)
- Python package updates:
  - jax from 0.3.23 to 0.3.25, jaxlib from 0.3.22 to 0.3.25
  - pyarrow from 6.0.1 to 9.0.0
  - torch from 1.12.1 to 1.13.0
  - torchaudio from 0.12.1 to 0.13.0
  - torchvision from 0.13.1 to 0.14.0
  - torchtext from 0.13.1 to 0.14.0
  - xlrd from 1.1.0 to 1.2.0
  - DriveFS from 62.0.1 to 66.0.3
- Made styling of markdown tables in outputs match markdown tables in text cells
- Improved formatting for empty interactive table rows
- Fixed syntax highlighting for variables with names that contain Python keywords (GitHub issue)

### 2022-11-11

- Added more dark editor themes for Monaco (when in dark mode, "Editor colorization" appears as an option in the Editor tab of the Tools → Settings dialog)
- Fixed bug where collapsed forms were deleted on mobile GitHub issue
- Python package updates:
  - rpy2 from 3.4.0 to 3.5.5 (GitHub issue)
  - notebook from 5.5.0 to 5.7.16
  - tornado from 5.1.1 to 6.0.4
  - tensorflow_probability from 0.16.0 to 0.17.0
  - pandas-gbq from 0.13.3 to 0.17.9
  - protobuf from 3.17.3 to 3.19.6
  - google-api-core[grpc] from 1.31.5 to 2.8.2
  - google-cloud-bigquery from 1.21.0 to 3.3.5
  - google-cloud-core from 1.0.1 to 2.3.2
  - google-cloud-datastore from 1.8.0 to 2.9.0
  - google-cloud-firestore from 1.7.0 to 2.7.2
  - google-cloud-language from 1.2.0 to 2.6.1
  - google-cloud-storage from 1.18.0 to 2.5.0
  - google-cloud-translate from 1.5.0 to 3.8.4

### 2022-10-21

- Launched a single-click way to get from BigQuery to Colab to further explore query results (announcement)
- Launched Pro, Pro+, and Pay As You Go to 19 additional countries: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, Greece, Hungary, Latvia, Lithuania, Norway, Portugal, Romania, Slovakia, Slovenia, and Sweden (tweet)




```
# reset index after removing some outliers and duplicates :-
df= df.reset_index(drop=True)
```

## b. Bivariate Analysis :

**In this section, I will analyze relationships between two variables and if possible, create new fwatures combining them for better visualization**

## 1. HbA1c_level vs. diabetes:

**HbA1c_level**

- A hemoglobin A1C (HbA1C) test is a blood test that shows what your average blood sugar (glucose) level was over the past two to three months.
- we will create a new feature bassed on the value of (HbA1C)

| HbA1c level | initial diagnosis |
|---|---|
| < 5.7 | Normal |
| 5.7 – 6.4 | Prediabetes |
| >= 6.5 | Diabetes |

**Resource :**

1. https://www.cdc.gov/diabetes/managing/managing-blood-sugar/a1c.html

```
diagnosing = []
for value in df['HbA1c_level']:
    if value < 5.7:
        diagnosing.append('normal')

    elif (5.7 <= value) and (value <= 6.4):
        diagnosing.append('prediabetes')

    elif (value >= 6.5):
        diagnosing.append('diabetes')

df['initial_diagnosis'] = diagnosing


# drow pie chart for each ['weight_type']
plt.figure(figsize = [14,8]);

plt.subplot(1,3,1);
plt.pie(df[df['initial_diagnosis'] == 'normal']['diabetes'].value_counts().values, label
        autopct='%1.2f%%',startangle = 45);
plt.title('initial diagnosis = normale');

plt.subplot(1,3,2);
plt.pie(df[df['initial_diagnosis'] == 'prediabetes']['diabetes'].value_counts().values,
       autopct='%1.2f%%',startangle = 45);
plt.title('initial diagnosis = prediabetes');
plt.legend();

plt.subplot(1,3,3);
plt.pie(df[df['initial_diagnosis'] == 'diabetes']['diabetes'].value_counts().values, lab
       autopct='%1.2f%%', startangle = 45);
plt.title('initial diagnosis = diabetes');
```

- Updated jax from 0.3.17 to 0.3.23, jaxlib from 0.3.15 to 0.3.22, TensorFlow from 2.8.2 to 2.9.2, CUDA from 11.1 to 11.2, and cuDNN from 8.0 to 8.1 (backend-info)
- Added a `readonly` option to `drive.mount`
- Fixed bug where Xarray was not working (GitHub issue)
- Modified Markdown parsing to ignore block quote symbol within MathJax (GitHub issue)

### 2022-09-30

- Launched Pay As You Go, allowing premium GPU access without requiring a subscription
- Added vim and tcllib to our runtime image
- Fixed bug where open files were closed on kernel disconnect (GitHub issue)
- Fixed bug where the play button/execution indicator was not clickable when scrolled into the cell output (GitHub issue)
- Updated the styling for form titles so that they avoid obscuring the code editor
- Created a GitHub repo, backend-info, with the latest apt-list.txt and pip-freeze.txt files for the Colab runtime (GitHub issue)
- Added `files.upload_file(filename)` to upload a file from the browser to the runtime with a specified filename

### 2022-09-16

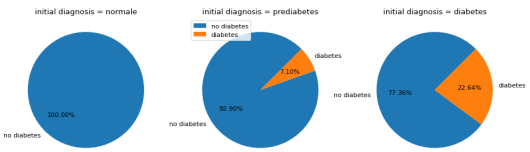
- Upgraded pymc from 3.11.0 to 4.1.4, jax from 0.3.14 to 0.3.17, jaxlib from 0.3.14 to 0.3.15, fsspec from 2022.8.1 to 2022.8.2
- Modified our save flow to avoid persisting Drive filenames as titles in notebook JSON
- Updated our Terms of Service
- Modified the `Jump to Cell` command to locate the cursor at the end of the command palette input (`Jump to cell` in Tools → Command palette in a notebook with section headings)
- Updated the styling of the Drive notebook comment UI
- Added support for terminating your runtime from code: `python from google.colab import runtime runtime.unassign()`
- Added regex filter support to the Recent notebooks dialog
- Inline google.colab.files.upload JS to fix `files.upload()` not working (GitHub issue)

### 2022-08-26

- Upgraded PyYAML from 3.13 to 6.0 (GitHub issue), drivefs from 61.0.3 to 62.0.1
- Upgraded TensorFlow from 2.8.2 to 2.9.1 and ipywidgets from 7.7.1 to 8.0.1 but rolled both back due to a number of user reports (GitHub issue, GitHub issue)
- Stop persisting inferred titles in notebook JSON (GitHub issue)
- Fix bug in background execution which affected some Pro+ users (GitHub issue)
- Fix bug where `Download as .py` incorrectly handled text cells ending in a double quote
- Fix bug for Pro and Pro+ users where we weren't honoring the preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Provide undo/redo for scratch cells
- When writing ipynb files, serialize empty multiline strings as `[]` for better consistency with JupyterLab

### 2022-08-11

- Upgraded ipython from 5.5.0 to 7.9.0, fbprophet 0.7 to prophet 1.1, tensorflow-datasets from 4.0.1 to 4.6.0, drivefs from 60.0.2 to 61.0.3, pytorch from 1.12.0 to 1.12.1, numba from 0.51 to 0.56, and lxml from 4.2.0 to 4.9.1
- Loosened our `requests` version requirement (GitHub issue)
- Removed support for TensorFlow 1
- Added Help → Report Drive abuse for Drive notebooks
- Fixed indentation for Python lines ending in [
- Modified styling of tables in Markdown to left-align them rather than centering them
- Fixed special character replacement when copying interactive tables as Markdown
- Fixed ansi 8-bit color parsing (GitHub issue)




- we can see that the value of **HbA1c_level** affect on the prediction whether it diabetes or not
- when the value of **HbA1c_level** lies on normale value , we see that there is no records with diabetes
- as **HbA1c_level** value increases, number of records with diabetes increases ## specially when it >= 6.5

| HbA1c level | initial diagnosis | actual diagnosis |
|---|---|---|
| < 5.7 | Normal | 100% no diabetes |
| 5.7 – 6.4 | Prediabetes | 7.47% have diabetes |
| >= 6.5 | Diabetes | 23.64% have diabetes |

## 2. bmi vs diabetes:

**BMI**

- Body mass index is a measure of body fat based on height and weight that applies to adult men and women.
- it is the weight in kilograms divided by height in meters squared

| BMI | Category |
|---|---|
| =< 18.5 | Underweight |
| 18.5 – 24.9 | Normal |
| 25 – 29.9 | Overweight |
| >= 30 | Obesity |

**Resources :**

1. https://my.clevelandclinic.org/health/diagnostics/12363-blood-glucose-test
2. https://www.cdc.gov/diabetes/basics/getting-tested.html

```
weight_type = []

for value in df['bmi']:
    if value <= 18.5:
        weight_type.append('underweight')

    elif (18.5 < value) and (value <= 24.9):
        weight_type.append('normal')

    elif (24.9 < value) and (value <= 29.9):
        weight_type.append('overweight')

    elif (value > 29.9):
        weight_type.append('obesity')

df['weight_type'] = weight_type


# drow pie chart for each ['weight_type']
plt.figure(figsize = [12,12]);

plt.subplot(2,2,1);
```

- Configured logging to preempt transitive imports and other loading from implicitly configuring the root logger
- Modified forms to use a value of `None` instead of causing a parse error when clearing raw and numeric-typed form fields

### 2022-07-22

- Update scipy from 1.4.1 to 1.7.3, drivefs from 59.0.3 to 60.0.2, pytorch from 1.11 to 1.12, jax & jaxlib from 0.3.8 to 0.3.14, opencv-python from 4.1.2.30 to 4.6.0.66, spaCy from 3.3.1 to 3.4.0, and dlib from 19.18.0 to 19.24.0
- Fix `Open in tab` doc link which was rendering incorrectly (GitHub issue)
- Add a preference for the default tab orientation to the Site section of the settings menu under Tools → Settings
- Show a warning for `USE_AUTH_EPHEM` usage when running authenticate_user on a TPU runtime (code)

### 2022-07-01

- Add a preference for code font to the settings menu under Tools → Settings
- Update drivefs from 58.0.3 to 59.0.3 and spacy from 2.2.4 to 3.3.1
- Allow display_data and execute_result text outputs to wrap, matching behavior of JupyterLab (does not affect stream outputs/print statements).
- Improve LSP handling of some magics, esp. %%writefile (GitHub issue).
- Add a FAQ entry about the mount Drive button behavior and include link buttons for each FAQ entry.
- Fix bug where the notebook was sometimes hidden behind other tabs on load when in single pane view.
- Fix issue with inconsistent scrolling when an editor is in multi-select mode.
- Fix bug where clicking on a link in a form would navigate away from the notebook
- Show a confirmation dialog before performing Replace all from the Find and replace pane.

### 2022-06-10

- Update drivefs from 57.0.5 to 58.0.3 and tensorflow from 2.8.0 to 2.8.2
- Support more than 100 repos in the GitHub repo selector shown in the open dialog and the clone to GitHub dialog
- Show full notebook names on hover in the open dialog
- Improve the color contrast for links, buttons, and the `ipywidgets.Accordion` widget in dark mode

### 2022-05-20

- Support URL params for linking to some common pref settings: force_theme=dark, force_corgi_mode=1, force_font_size=14. Params forced by URL are not persisted unless saved using Tools → Settings.
- Add a class `markdown-google-sans` to allow Markdown to render in Google Sans
- Update monaco-vim from 0.1.19 to 0.3.4
- Update drivefs from 55.0.3 to 57.0.5, jax from 0.3.4 to 0.3.8, and jaxlib from 0.3.2 to 0.3.7

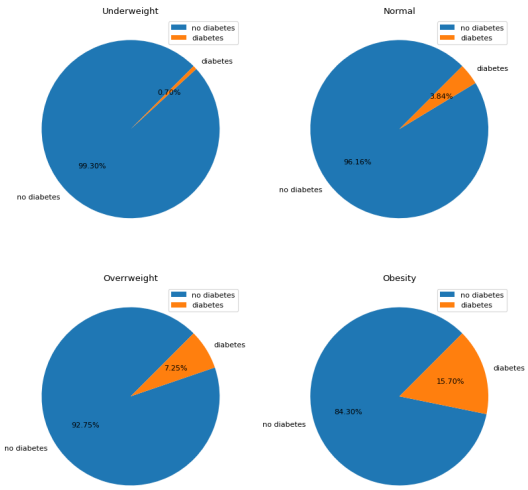
### 2022-04-29

- Added mode (under Miscellaneous in Tools → Settings)
- Added "Disconnect and delete runtime" option to the menu next to the Connect button
- Improved rendering of filter options in an interactive table
- Added git-lfs to the base image
- Updated torch from 1.10.0 to 1.11.0, jupyter-core from 4.9.2 to 4.10.0, and cmake from 3.12.0 to 3.22.3
- Added more details to our FAQ about unsupported uses (using proxies, downloading torrents, etc.)
- Fixed issue with apt-get dependencies

### 2022-04-15

- Add an option in the file browser to show hidden files.
- Upgrade gdown from 4.2.0 to 4.4.0, google-api-core[grpc] from 1.26.0 to 1.31.5, and pytz from 2018.4 to 2022.1

```
plt.pie(df[df['weight_type'] == 'underweight']['diabetes'].value_counts().values, labels
       autopct='%1.2f%%', startangle = 45);
plt.title('Underweight')
plt.legend();

plt.subplot(2,2,2);
plt.pie(df[df['weight_type'] == 'normal']['diabetes'].value_counts().values, labels = ['
       autopct='%1.2f%%',startangle = 45);
plt.title('Normal')
plt.legend();

plt.subplot(2,2,3);
plt.pie(df[df['weight_type'] == 'overweight']['diabetes'].value_counts().values, labels
       autopct='%1.2f%%',startangle = 45);
plt.title('Overrweight')
plt.legend();

plt.subplot(2,2,4);
plt.pie(df[df['weight_type'] == 'obesity']['diabetes'].value_counts().values, labels = [
       autopct='%1.2f%%',startangle = 45);
plt.title('Obesity')
plt.legend();
```




- according to ordinal category **['underweight','normal','overweight','obesity']**, as weight category increases, percentage of patients with diabetes increase.

| BMI | Category | Prediction |
|---|---|---|
| =< 18.5 | Underweight | 0.7% have diabetes |
| 18.5 – 24.9 | Normal | 3.85% have diabetes |
| 25 – 29.9 | Overweight | 7.88% have diabetes |
| >= 30 | Obesity | 15.94% have diabetes |

### 2022-03-25

- Launched Pro/Pro+ to 12 additional countries: Australia, Bangladesh, Colombia, Hong Kong, Indonesia, Mexico, New Zealand, Pakistan, Philippines, Singapore, Taiwan, and Vietnam
- Added `google.colab.auth.authenticate_service_account(` to support using Service Account keys
- Update jax from 0.3.1 to 0.3.4 & jaxlib from 0.3.0 to 0.3.2
- Fixed an issue with Twitter previews of notebooks shared as Github Gists

### 2022-03-10

- Launched Pro/Pro+ to 10 new countries: Ireland, Israel, Italy, Morocco, the Netherlands, Poland, Spain, Switzerland, Turkey, and the United Arab Emirates
- Launched support for scheduling notebooks for Pro+ users
- Fixed bug in interactive datatables where filtering by number did not work
- Finished removing the python2 kernelspec

### 2022-02-25

- Made various accessibility improvements to the header
- Fix bug with forms run:auto where a form field change would trigger multiple runs
- Minor updates to the bigquery example notebook and snippet
- Include background execution setting in the sessions dialog for Pro+ users
- Update tensorflow-probability from 0.15 to 0.16
- Update jax from 0.2.25 to 0.3.1 & jaxlib from 0.1.71 to 0.3.0

### 2022-02-11

- Improve keyboard navigation for the open dialog
- Fix issue where nvidia-smi stopped reporting resource utilization for some users who were modifying the version of nvidia used
- Update tensorflow from 2.7 to 2.8, keras from 2.7 to 2.8, numpy from 1.19.5 to 1.21.5, tables from 3.4.4 to 3.7.0

### 2022-02-04

- Improve UX for opening content alongside your notebook, such as files opened from the file browser. This includes a multi-pane view and drag-drop support
- Better Twitter previews when sharing example Colab notebooks and notebooks opened from GitHub Gists
- Update pandas from 1.1.5 to 1.3.5
- Update openpyxl from 2.5.9 to 3.0.0 and pyarrow from 3.0.0 to 6.0.0
- Link to the release notes from the Help menu

### 2022-01-28

- Add a copy button to data tables
- Python LSP support for better completions and code diagnostics. This can be configured in the Editor Settings (Tools → Settings)
- Update gspread examples in our documentation
- Update gdown from 3.6 to 4.2

### 2022-01-21

- New documentation for the `google.colab` package
- Show GPU RAM in the resource usage tab
- Improved security for mounting Google Drive which disallows mounting Drive from accounts other than the one currently executing the notebook

### 2022-01-14

- Add a preference (Tools → Settings) to use a temporary scratch notebook as the default landing page
- Fix bug where / and : weren't working in VIM mode
- Update gspread from 3.0 to 3.4
- Update the Colab Marketplace VM image

## 3. blood_glucose_level vs diabetes :-

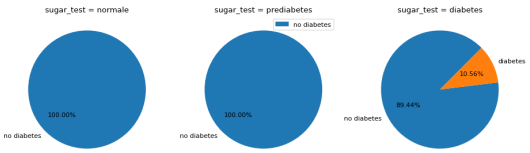
**blood_glucose_level**

| blood_glucose_level | Category |
|---|---|
| =< 99 | normal |
| 100 – 125 | Prediabetes |
| >= 126 | Diabetes |

**Resources :**

1. https://my.clevelandclinic.org/health/diagnostics/12363-blood-glucose-test
2. https://www.cdc.gov/diabetes/basics/getting-tested.html

```
sugar_test = []

for value in df['blood_glucose_level']:
    if value <= 99:
        sugar_test.append('normal')

    elif (99 < value) and (value <= 125):
        sugar_test.append('prediabetes')

    elif (value > 125):
        sugar_test.append('diabetes')

df['sugar_test'] = sugar_test


# drow pie chart for each ['weight_type']
plt.figure(figsize = [14,8]);

plt.subplot(1,3,1);
plt.pie(df[df['sugar_test'] == 'normal']['diabetes'].value_counts().values, labels = ['n
        autopct='%1.2f%%',startangle = 45);
plt.title('sugar_test = normale');

plt.subplot(1,3,2);
plt.pie(df[df['sugar_test'] == 'prediabetes']['diabetes'].value_counts().values, labels
       autopct='%1.2f%%',startangle = 45);
plt.title('sugar_test = prediabetes');
plt.legend();

plt.subplot(1,3,3);
plt.pie(df[df['sugar_test'] == 'diabetes']['diabetes'].value_counts().values, labels = [
       autopct='%1.2f%%', startangle = 45);
plt.title('sugar_test = diabetes');
```



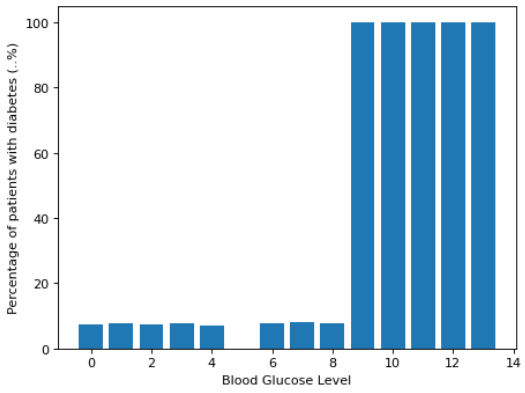

## all patients with diabetes have high sugar test result ( > 126)

```
# deep exploration :....

# isolate record with 'blood_glucose_level' values >= 126.
high_sugar_result = df[df['blood_glucose_level'] >= 126]

# counts number of patients have diabetes.
```

```
counts  = pd.DataFrame(high_sugar_result.groupby(['blood_glucose_level'])['diabetes'].su

# create a columns with total number of patients and the percentage of them who have dia
counts['total'] = high_sugar_result['blood_glucose_level'].value_counts().sort_index()
counts['percentage'] = round((counts['diabetes'] / counts['total'])*100,2)

counts = counts.reset_index()
counts
```

| | blood_glucose_level | diabetes | total | percentage |
|---|---|---|---|---|
| **0** | 126 | 527 | 7190 | 7.33 |
| **1** | 130 | 566 | 7231 | 7.83 |
| **2** | 140 | 522 | 7178 | 7.27 |
| **3** | 145 | 543 | 7142 | 7.60 |
| **4** | 155 | 482 | 7019 | 6.87 |
| **5** | 158 | 0 | 6599 | 0.00 |
| **6** | 159 | 544 | 7197 | 7.56 |
| **7** | 160 | 584 | 7150 | 8.17 |
| **8** | 200 | 542 | 7081 | 7.65 |
| **9** | 220 | 500 | 500 | 100.00 |
| **10** | 240 | 537 | 537 | 100.00 |
| **11** | 260 | 518 | 518 | 100.00 |
| **12** | 280 | 601 | 601 | 100.00 |
| **13** | 300 | 556 | 556 | 100.00 |

```
# barplot for visual representation :
plt.bar(counts.index, counts['percentage']);
plt.xlabel('Blood Glucose Level');
plt.ylabel('Percentage of patients with diabetes (..%)');
```




- Patients with **['blood_glucose_level']** between **126 to 200** , around 7.5 of them have already diabetes and that whose **['blood_glucose_level'] >= 220** , all of then have accually diabetes .

## # 5. Train-test split and standardized scaler

```
## Converting categorical to Numerical. One Hot Encoding
df = pd.get_dummies(df)


from sklearn.model_selection import train_test_split

X = df.drop('diabetes',axis=1)
y = df['diabetes']
```

```
y = df['diabetes']

X_train,X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


from sklearn.preprocessing import StandardScaler

scale = StandardScaler()
X_train=scale.fit_transform(X_train)
X_test=scale.transform(X_test)


print("X_train:", X_train.shape, "\ny_train:", y_train.shape)
print("X_test:", X_test.shape, "\ny_test:", y_test.shape)
```

```
X_train: (74317, 18)
y_train: (74317,)
X_test: (18580, 18)
y_test: (18580,)
```

## Logistic Regression Classifer

```
## Designing Logistic Regeression Classifier
LogReg = LogisticRegression()
LogReg.fit(X_train, y_train)
LogReg_pred = LogReg.predict(X_test)
LogReg_acc = accuracy_score(y_test, LogReg_pred)
LogReg_mae = mean_absolute_error(y_test, LogReg_pred)
LogReg_mse = mean_squared_error(y_test, LogReg_pred)
LogReg_rmse = np.sqrt(mean_squared_error(y_test, LogReg_pred))
LogReg_precision = precision_score(y_test, LogReg_pred)
LogReg_recall = recall_score(y_test, LogReg_pred)
LogReg_f1 = f1_score(y_test, LogReg_pred)


## Printing the results

print("The accuracy for Logistic Regression is", LogReg_acc)
print("The classification report using Logistic Regression is:")
print(classification_report(y_test, LogReg_pred))
```

```
The accuracy for Logistic Regression is 0.9634553283100108
The classification report using Logistic Regression is:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     17183
           1       0.86      0.61      0.71      1397

    accuracy                           0.96     18580
   macro avg       0.92      0.80      0.85     18580
weighted avg       0.96      0.96      0.96     18580
```

```
## Confusion Matrix
LogReg_cm = confusion_matrix(y_test, LogReg_pred)
sns.heatmap(LogReg_cm/np.sum(LogReg_cm), annot = True, fmt = '0.2%', cmap = 'Oranges')
plt.title("Logisitic Regression Confusion Matrix")
```

```
Text(0.5, 1.0, 'Logisitic Regression Confusion Matrix')
```

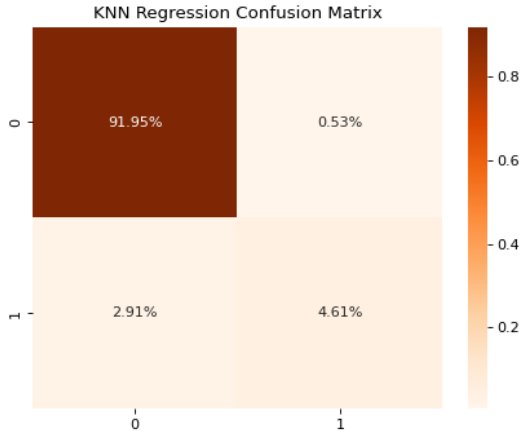

## K-Nearest Neighbour Regression Classifier

```
## Designing KNN Classifier

KNN = KNeighborsClassifier()
KNN.fit(X_train, y_train)
KNN_pred = KNN.predict(X_test)
KNN_acc = accuracy_score(y_test, KNN_pred)
KNN_mae = mean_absolute_error(y_test, KNN_pred)
KNN_mse = mean_squared_error(y_test, KNN_pred)
KNN_rmse = np.sqrt(mean_squared_error(y_test, KNN_pred))
KNN_precision = precision_score(y_test, KNN_pred)
KNN_recall = recall_score(y_test, KNN_pred)
KNN_f1 = f1_score(y_test, KNN_pred)


##Printing the results
print("The accuracy for KNeighbors is", KNN_acc)
print("The classification report using KNeighbors is:" ),
print(classification_report(y_test, KNN_pred))
```

```
The accuracy for KNeighbors is 0.9655543595263725
The classification report using KNeighbors is:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     17183
           1       0.90      0.61      0.73      1397

    accuracy                           0.97     18580
   macro avg       0.93      0.80      0.85     18580
weighted avg       0.96      0.97      0.96     18580
```

```
##Confusion Matrix
KNN_cm = confusion_matrix(y_test, KNN_pred)
sns.heatmap(KNN_cm/np.sum(KNN_cm), annot = True, fmt = '0.2%', cmap = 'Oranges')
plt.title("KNN Regression Confusion Matrix")
```

```
Text(0.5, 1.0, 'KNN Regression Confusion Matrix')
```




## Decision Tree Classifier

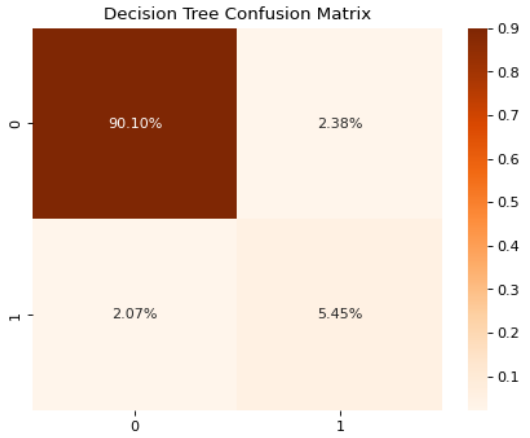
```
##Designing Decision Tree Classifier
DecTree = DecisionTreeClassifier()
DecTree.fit(X_train, y_train)
DecTree_pred = DecTree.predict(X_test)
DecTree_acc = accuracy_score(y_test, DecTree_pred)
DecTree_precision = precision_score(y_test, DecTree_pred)
DecTree_recall = recall_score(y_test, DecTree_pred)
DecTree_f1 = f1_score(y_test, DecTree_pred)
```

```
##Printing the results
print("The accuracy for Decision Tree is", DecTree_acc)
print("The classification report using Decision Tree is:")
print(classification_report(y_test, DecTree_pred))
```

```
The accuracy for Decision Tree is 0.9554359526372443
The classification report using Decision Tree is:
              precision    recall  f1-score   support

           0       0.98      0.97      0.98     17183
           1       0.70      0.72      0.71      1397

    accuracy                           0.96     18580
   macro avg       0.84      0.85      0.84     18580
weighted avg       0.96      0.96      0.96     18580
```

```
##Confusion Matrix
DecTree_cm = confusion_matrix(y_test, DecTree_pred)
sns.heatmap(DecTree_cm/np.sum(DecTree_cm), annot = True, fmt = '0.2%', cmap = 'Oranges')
plt.title("Decision Tree Confusion Matrix")
```

```
Text(0.5, 1.0, 'Decision Tree Confusion Matrix')
```




## Random Forest Classifier

```
##Designing Randoom Forest Classifier
RFTree = RandomForestClassifier()
RFTree.fit(X_train, y_train)
RFTree_pred = RFTree.predict(X_test)
RFTree_acc = accuracy_score(y_test, RFTree_pred)
RFTree_precision = precision_score(y_test, RFTree_pred)
RFTree_recall = recall_score(y_test, RFTree_pred)
RFTree_f1 = f1_score(y_test, RFTree_pred)
```

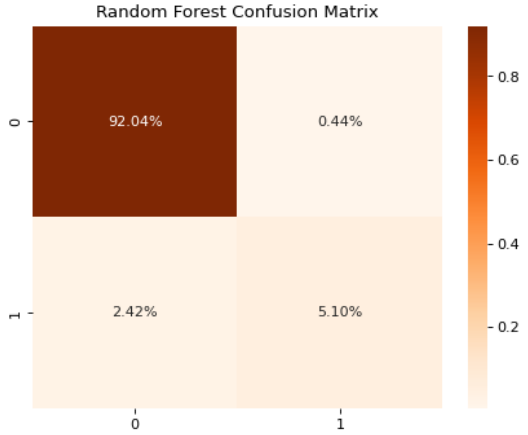
```
##Printing the results
print("The accuracy for Random Forest is", RFTree_acc)
print("The classification report using Random Forest is:")
print(classification_report(y_test, RFTree_pred))
```

```
The accuracy for Random Forest is 0.9713670613562971
The classification report using Random Forest is:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98     17183
           1       0.92      0.68      0.78      1397

    accuracy                           0.97     18580
   macro avg       0.95      0.84      0.88     18580
weighted avg       0.97      0.97      0.97     18580
```

```
##Confusion Matrix
RFTree_cm = confusion_matrix(y_test, RFTree_pred)
```

```
sns.heatmap(RFTree_cm/np.sum(RFTree_cm), annot = True, fmt = '0.2%', cmap = 'Oranges')
plt.title("Random Forest Confusion Matrix")
```

```
Text(0.5, 1.0, 'Random Forest Confusion Matrix')
```




## Support Vector Machine Classifier

```
#Designing SVM Classifier
SVM = SVC()
SVM.fit(X_train, y_train)
SVM_pred = SVM.predict(X_test)
SVM_acc = accuracy_score(y_test, SVM_pred)
SVM_precision = precision_score(y_test, SVM_pred)
SVM_recall = recall_score(y_test, SVM_pred)
SVM_f1 = f1_score(y_test, SVM_pred)
```

```
##Pritning the results
print("The accuracy for SVM is", SVM_acc)
print("The classification report using SVM is:", SVM_acc)
print(classification_report(y_test, SVM_pred))
```

```
The accuracy for SVM is 0.9695371367061356
The classification report using SVM is: 0.9695371367061356
              precision    recall  f1-score   support

           0       0.97      1.00      0.98     17183
           1       0.99      0.60      0.75      1397

    accuracy                           0.97     18580
   macro avg       0.98      0.80      0.87     18580
weighted avg       0.97      0.97      0.97     18580
```

```
##Confusion Matrix
SVM_cm = confusion_matrix(y_test, SVM_pred)
sns.heatmap(SVM_cm/np.sum(SVM_cm), annot = True, fmt = '0.2%', cmap = 'Oranges')
plt.title("SVM Confusion Matrix")
```

```
Text(0.5, 1.0, 'SVM Confusion Matrix')
```




## Comparison of Different Models



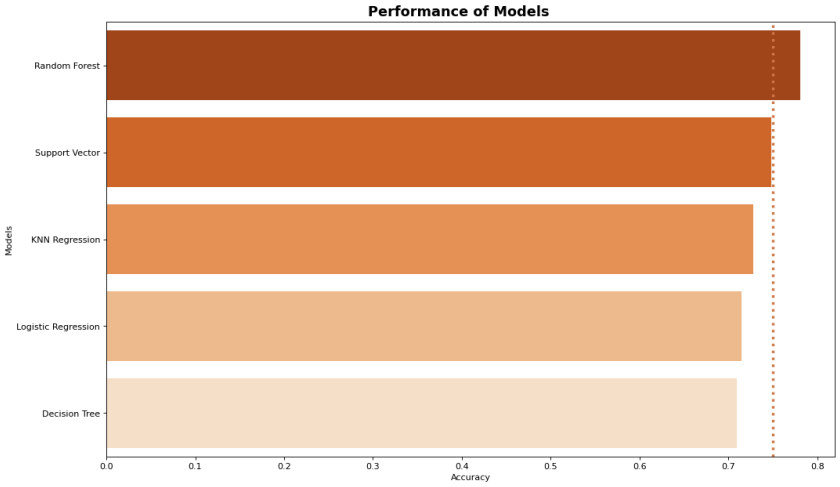

```
models = pd.DataFrame({
    'Model':['Logistic Regression', 'KNN Regression', 'Decision Tree', 'Random Forest',
    'Accuracy' :[LogReg_acc, KNN_acc, DecTree_acc, RFTree_acc, SVM_acc],
    'Precision' :[LogReg_precision, KNN_precision, DecTree_precision, RFTree_precision,
    'Recall' :[LogReg_recall, KNN_recall, DecTree_recall, RFTree_recall, SVM_recall],
    'F1 Score' :[LogReg_f1, KNN_f1, DecTree_f1, RFTree_f1, SVM_f1]
})
models = models.sort_values(by='Accuracy', ascending=False)
```

```
models
```

| | Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| **3** | Random Forest | 0.971367 | 0.920311 | 0.677881 | 0.780709 |
| **4** | Support Vector | 0.969537 | 0.985965 | 0.603436 | 0.748668 |
| **1** | KNN Regression | 0.965554 | 0.896335 | 0.612742 | 0.727891 |
| **0** | Logistic Regression | 0.963455 | 0.864837 | 0.609162 | 0.714826 |
| **2** | Decision Tree | 0.955436 | 0.695533 | 0.724409 | 0.709677 |

```
fig = plt.figure(figsize=(15,9))

ax = sns.barplot(data=models,
                 y='Model',
                 x='Accuracy',
                 palette = 'Blues_r')

ax.figure.suptitle('Performance of Models',y=0.91, size = 16, color = 'black', weight='b

plt.xlabel('Accuracy')
plt.ylabel('Models')
plt.axvline(x = 0.96, ymin = 0, ymax = 1,
            linewidth = 3, linestyle =":",
            color ='#cf7849');
```

```
fig = plt.figure(figsize=(15,9))

ax = sns.barplot(data=models,
                 y='Model',
                 x='F1 Score',
                 palette = 'Oranges_r')

ax.figure.suptitle('Performance of Models',y=0.91, size = 16, color = 'black', weight='b

plt.xlabel('Accuracy')
plt.ylabel('Models')
plt.axvline(x = 0.75, ymin = 0, ymax = 1,
            linewidth = 3, linestyle =":",
            color ='#cf7849');
```

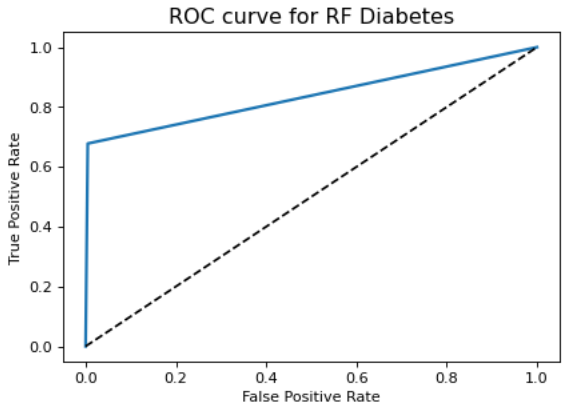



## CROSS VALIDATION

```
##Performing K-Fold cross Validation

from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from statistics import mean, stdev

cv = KFold(n_splits=10, random_state=1, shuffle=True)

KNN_scores = cross_val_score(KNN, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
DecTree_scores = cross_val_score(DecTree, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
RFTree_scores = cross_val_score(RFTree, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
SVM_scores = cross_val_score(SVM, X, y, scoring='accuracy', cv=cv, n_jobs=-1)

print('Accuracy of CV - KNN: %.4f (%.4f)' % (mean(KNN_scores), stdev(KNN_scores)))
print('Accuracy of CV - Decision Tree: %.4f (%.4f)' % (mean(DecTree_scores), stdev(DecTr
print('Accuracy of CV - Random Forest: %.4f (%.4f)' % (mean(RFTree_scores), stdev(RFTree
print('Accuracy of CV - SVM: %.4f (%.4f)' % (mean(SVM_scores), stdev(SVM_scores)))
```

```
Accuracy of CV - KNN: 0.9605 (0.0021)
Accuracy of CV - Decision Tree: 0.9573 (0.0029)
Accuracy of CV - Random Forest: 0.9716 (0.0018)
Accuracy of CV - SVM: 0.9514 (0.0034)
```

```
## Plotting ROC Curve for best performing Model (RF Classifier)

from sklearn.metrics import roc_curve

fpr, tpr, thresholds = roc_curve(y_test, RFTree_pred)

plt.figure(figsize=(6,4))
plt.plot(fpr, tpr, linewidth=2)
plt.plot([0,1], [0,1], 'k--' )
plt.rcParams['font.size'] = 12
plt.title('ROC curve for RF Diabetes')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```




```
from sklearn.metrics import roc_auc_score

ROC_AUC = roc_auc_score(y_test, RFTree_pred)
print('ROC AUC : {:.4f}'.format(ROC_AUC))
```

```
ROC AUC : 0.8366
```

```
Cross_validated_ROC_AUC = cross_val_score(RFTree, X_train, y_train, cv=10, scoring='roc_

print('Cross validated ROC AUC : {:.4f}'.format(Cross_validated_ROC_AUC))
```

```
Cross validated ROC AUC : 0.9572
```

It is eveident that ROC AUC is better after the cross-validation.So, the cross-validation improved the model.