Minor project on

CHURN PREDICTION USING ANN

INTERNSHIP ON

ARTIFICIAL INTELLIGENCE WITH PYTHON

ORGANISED BY : DEXTERITY EDTECH PVT.LTD

Submitted by-

Kumari Nikita

Introduction to Credit Card Churn Prediction Using ANN Models

In the competitive financial services industry, credit card issuers face a constant challenge of retaining their customer base. Customer churn, or the attrition of credit card users, can significantly impact a company’s profitability and long-term sustainability. Predicting and mitigating churn is thus essential for credit card companies to maintain their market position and ensure customer loyalty. Artificial Neural Networks (ANNs) have emerged as a powerful tool in the realm of predictive analytics, offering a sophisticated approach to identifying customers at risk of discontinuing their credit card services.

ANNs are inspired by the structure and functionality of the human brain, comprising layers of interconnected neurons that process and analyze data in a non-linear fashion. This capability makes ANNs particularly well-suited for handling the complexity and variability inherent in customer behaviour data. By leveraging historical data on credit card usage, transaction patterns, payment history, customer demographics, and other relevant features, ANNs can uncover intricate patterns that traditional statistical methods might fail to detect.

The process of developing a churn prediction model for credit card customers using ANNs begins with data collection and preprocessing. High-quality data is crucial for model accuracy, necessitating steps such as data cleaning, normalization, and feature engineering. The dataset is then divided into training and testing subsets, enabling the ANN to learn from past data and validate its predictions on new, unseen instances. Designing the ANN’s architecture involves determining the optimal number of layers, neurons per layer, and activation functions to balance model complexity and computational efficiency.

Once trained, the ANN model undergoes rigorous evaluation using metrics such as accuracy, precision, recall, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics provide insights into the model’s ability to correctly identify customers who are likely to churn, as well as those who will remain loyal. Techniques such as cross-validation and hyperparameter tuning are employed to refine the model and enhance its generalization capability.

Implementing ANN-based churn prediction models for credit card customers offers several advantages. Firstly, the ability of ANNs to process large volumes of data and detect non-linear relationships enhances the accuracy of predictions. Secondly, ANNs are adaptable and can be retrained with new data to reflect changes in customer behaviour and market conditions, ensuring sustained relevance and performance. The insights generated by the model can inform targeted retention strategies, such as personalized marketing campaigns, loyalty programs, and proactive customer support interventions.

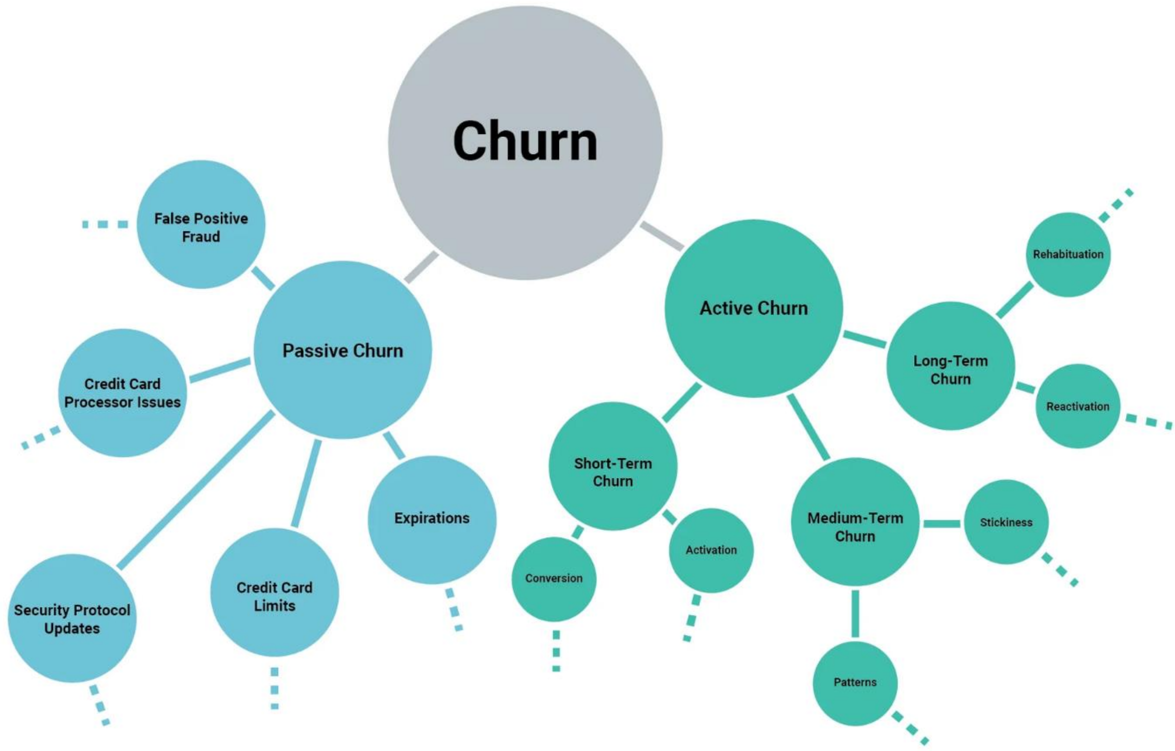
Moreover, by predicting churn, credit card companies can allocate resources more effectively and prioritize efforts to retain high-value customers. The adoption of ANN models also fosters a data-driven decision-making culture within organizations, encouraging continuous improvement and innovation in customer relationship management

In conclusion, the use of Artificial Neural Networks for predicting credit card customer churn represents a cutting-edge approach in the financial services sector. By harnessing the predictive power of ANNs, credit card issuers can proactively identify at-risk customers, tailor interventions to meet their needs, and ultimately reduce churn rates. As the industry continues to evolve, the integration of ANN-based churn prediction models will be instrumental in fostering customer loyalty and driving long-term business success.

WHAT IS CHURN?

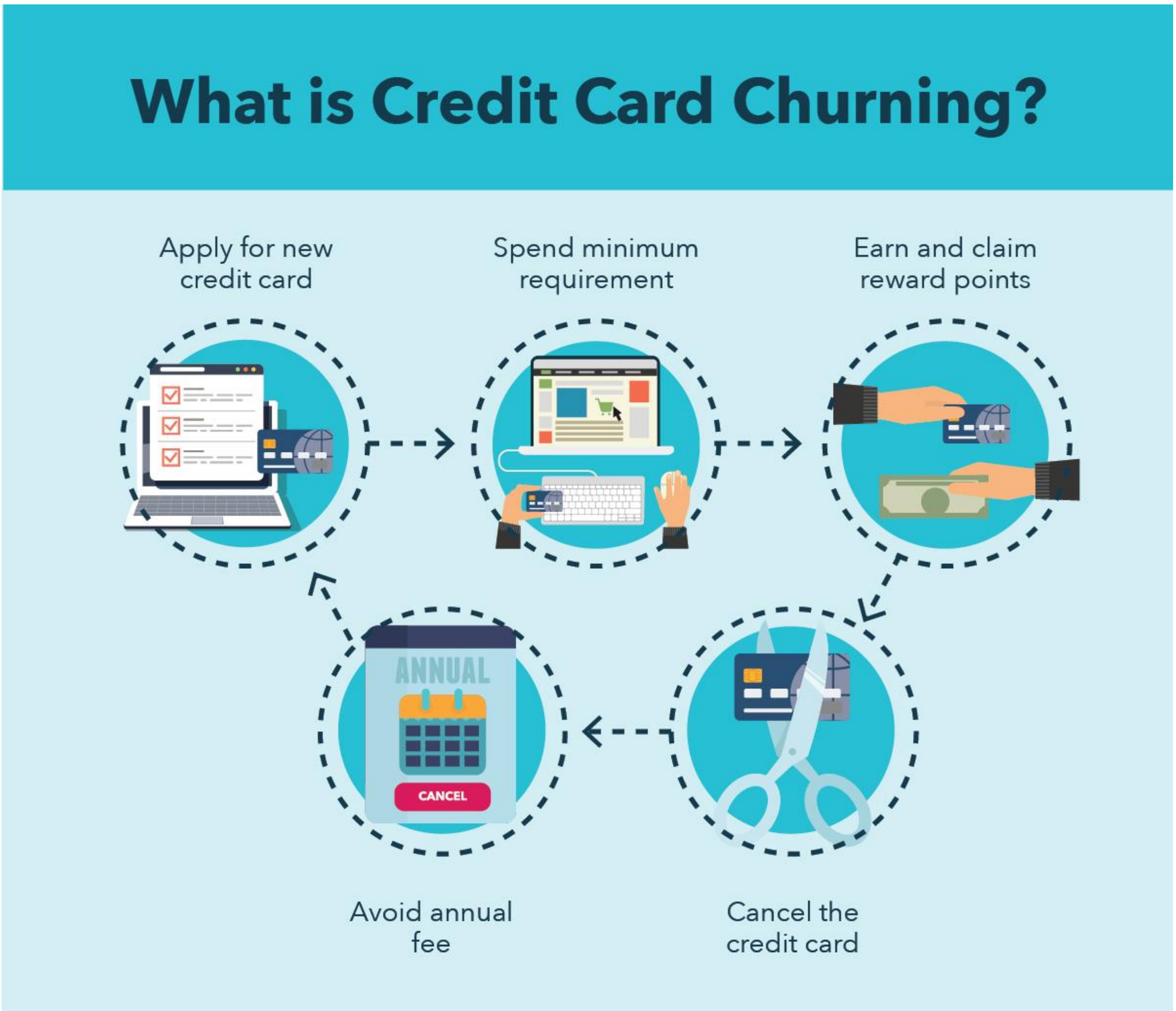
Churn, in a business context, refers to the phenomenon of customers discontinuing their relationship with a company or ceasing to use its products or services. It is a critical metric for businesses as it directly impacts their revenue and growth. Customer churn can be measured as the percentage of customers who leave over a specific period.

Understanding and predicting churn is essential for businesses to retain customers and maintain a competitive edge. By identifying factors that contribute to churn, companies can implement strategies to improve customer satisfaction, enhance loyalty, and reduce attrition rates.



ABSTRACT

Credit card churn prediction has become a critical task for financial institutions striving to retain their customer base and maintain profitability. This paper explores the application of Artificial Neural Networks (ANNs) for predicting credit card customer churn, leveraging their capacity to model complex, non-linear relationships within customer data. By analysing historical data encompassing transaction patterns, payment behaviours, and demographic information, ANNs can accurately identify customers at risk of discontinuing their credit card services. The study outlines the data preprocessing steps, model architecture design, and evaluation metrics employed to develop a robust churn prediction model. The results demonstrate that ANNs offer superior predictive accuracy compared to traditional methods, enabling credit card issuers to implement targeted retention strategies. The implications of this study suggest that integrating ANN-based models into customer relationship management systems can significantly enhance retention efforts and drive long-term business success.



DATASET

A dataset for credit card churn prediction typically includes a variety of features that capture customer demographics, transaction behaviours, and account details.

1. **Customer Demographics**: Age, gender, marital status, education level, and income category.
2. **Account Information**: Account age, credit limit, and total relationship count (number of products held by the customer).
3. **Transaction History**: Total transaction amount and count over a specific period (e.g., last 12 months), average utilization ratio (how much of the available credit is used), and changes in transaction amounts between quarters.
4. **Behavioural Metrics**: Number of months inactive, number of contacts with the bank in the last 12 months, and total revolving balance (balance carried over from one month to the next).
5. **Attrition Flag**: Whether the customer has left the credit card service or not.

These features help in building a predictive model to identify customers who are at risk of churning, allowing the bank to implement targeted retention strategies.

LIBRARIES USED

➢ TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

➢ Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

- ## Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

- ## Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

- ## Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

MACHINE LEARNING MODELS

Artificial Neural Networks (ANNs)-

Artificial Neural Networks (ANNs) are a type of machine learning model inspired by the biological neural networks in the human brain. Here's a closer look at what makes ANNs tick:

Structure

- **Neurons and Layers**: ANNs consist of interconnected layers of artificial neurons. There are typically three types of layers:
    - **Input Layer**: Receives the initial data.
    - **Hidden Layers**: Perform complex computations. There can be multiple hidden layers in deep neural networks.
    - **Output Layer**: Produces the final prediction or classification.

Functionality

- **Activation Functions**: These functions determine the output of each neuron. Common activation functions include the Sigmoid, Tanh, and ReLU (Rectified Linear Unit).
- **Weights and Biases**: Each connection between neurons has an associated weight, which is adjusted during training to minimize prediction error. Biases are additional parameters that help the model make more accurate predictions.
- **Forward and Backward Propagation**:
    - **Forward Propagation**: Input data is passed through the network layer by layer to generate an output.
    - **Backward Propagation**: The network adjusts weights and biases based on the error between the predicted output and the actual output, using techniques like gradient descent.

Training Process

- The training process of an ANN involves forward and backward propagation. During forward propagation, input data passes through the network to generate an output. The network’s predictions are then compared to actual values, and the error is calculated. In backward propagation, this error is propagated back through the network, adjusting the weights and biases using algorithms like gradient descent. This iterative process continues until the model converges to an optimal set of parameters that minimize prediction error. The robustness of an ANN heavily depends on the quality and quantity of the training data, as well as the architecture design, including the number of layers and neurons, and the choice of hyperparameters.

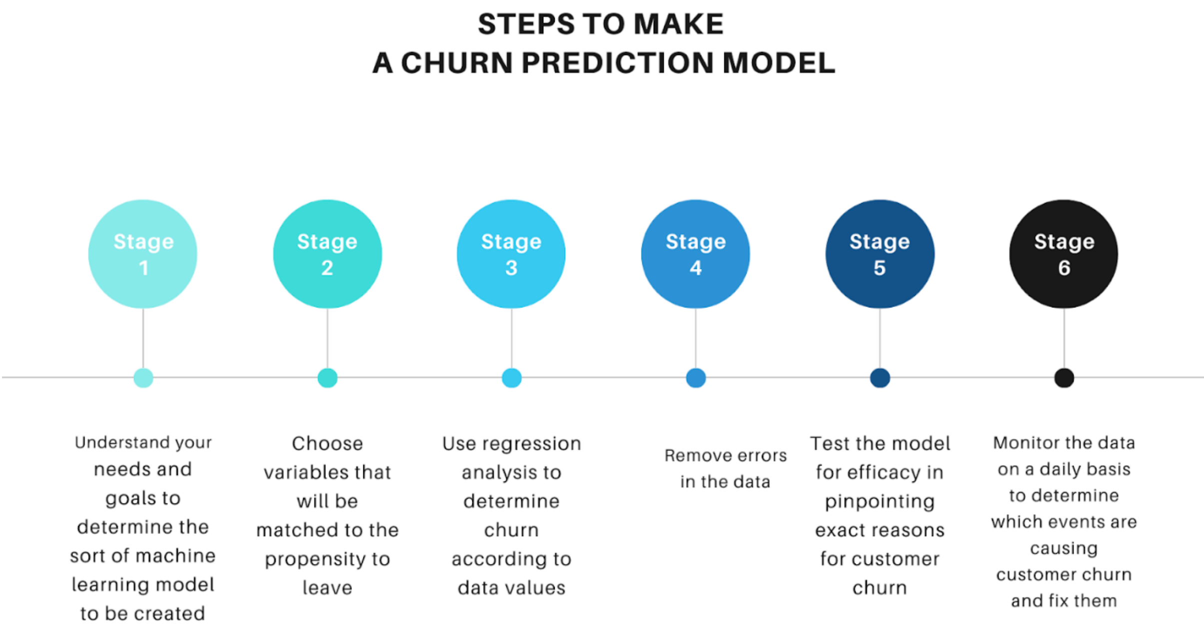
- ANNs have demonstrated remarkable success across a wide range of applications. In image and speech recognition, ANNs can identify patterns and features that are often imperceptible to human eyes and ears. In natural language processing, they facilitate tasks such as translation, sentiment analysis, and chatbot interactions. In finance, ANNs predict stock prices, assess credit risks, and detect fraudulent transactions. The versatility of ANNs extends to healthcare, where they assist in diagnosing diseases, predicting patient outcomes, and personalizing treatment plans.

- Despite their capabilities, ANNs pose certain challenges. They require extensive computational resources, particularly for deep networks with numerous layers and neurons. Training deep ANNs can be time-consuming and resource-intensive, often necessitating specialized hardware such as GPUs. Additionally, ANNs are sometimes criticized for being "black boxes" because it can be difficult to interpret how they arrive at specific decisions. This lack of interpretability can be a drawback in applications where understanding the decision-making process is crucial.

- Recent advancements in neural network research have led to the development of more sophisticated architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs are particularly effective for image processing tasks, leveraging convolutional layers to capture spatial hierarchies in data. RNNs, on the other hand, are well-suited for sequential data analysis, excelling in tasks such as language modeling and time series prediction. These specialized neural networks have further expanded the horizons of what ANNs can achieve.

- In summary, Artificial Neural Networks are a foundational element of modern machine learning, offering unparalleled capabilities for modeling complex data patterns. Their applications span diverse domains, from computer vision and natural language processing to finance and healthcare. While they present challenges related to computational demands and interpretability, ongoing research and technological advancements continue to enhance their performance and accessibility. As the field of machine learning evolves, ANNs will undoubtedly remain at the forefront, driving innovation and unlocking new possibilities for data-driven decision-making.

CODE



STEPS TO MAKE
A CHURN PREDICTION MODEL
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage 6
Understand your needs and goals to determine the sort of machine learning model to be created
Choose variables that will be matched to the propensity to leave
Use regression analysis to determine churn according to data values
Remove errors in the data
Test the model for efficacy in pinpointing exact reasons for customer churn
Monitor the data on a daily basis to determine which events are causing customer churn and fix them

By Kumari Nikita

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

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

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & 
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/credit-card-customer-churn-prediction/Churn_Modelling.csv
```

```
df = pd.read_csv('/kaggle/input/credit-card-customer-churn-prediction/Churn_Modelling.csv')
```

```
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Esti
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	

```
df.drop(columns = ['RowNumber','CustomerId','Surname'],inplace=True)
```

```
df.head()
```

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

```
df['Geography'].value_counts()
```

```
Geography
France     5014
Germany    2509
Spain      2477
Name: count, dtype: int64
```

```
df['Gender'].value_counts()
```

```
Gender
Male      5457
Female    4543
Name: count, dtype: int64
```

```
df = pd.get_dummies(df,columns=['Geography','Gender'],drop_first=True)
```

```
df.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germany	Geography_S
0	619	42	2	0.00	1	1	1	101348.88	1	False	F
1	608	41	1	83807.86	1	0	1	112542.58	0	False	
2	502	42	8	159660.80	3	1	0	113931.57	1	False	F
3	699	39	1	0.00	2	0	0	93826.63	0	False	F
4	850	43	2	125510.82	1	1	1	79084.10	0	False	

```
X = df.drop(columns=['Exited'])
y = df['Exited'].values

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

X_train_trf = scaler.fit_transform(X_train)
X_test_trf = scaler.transform(X_test)
```

```
import tensorflow
from tensorflow import keras
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
```

```
model = Sequential()

model.add(Dense(11,activation='sigmoid',input_dim=11))
model.add(Dense(11,activation='sigmoid'))
model.add(Dense(11,activation='sigmoid'))
model.add(Dense(1,activation='sigmoid'))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argumen
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
model.summary()
```

Model: "sequential_9"

Layer (type)	Output Shape	Param #
dense_21 (Dense)	(None, 11)	132
dense_22 (Dense)	(None, 11)	132
dense_23 (Dense)	(None, 11)	132
dense_24 (Dense)	(None, 1)	12

Total params: 408 (1.59 KB)

```
model.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['accuracy'])
```

```
history = model.fit(X_train,y_train,batch_size=50,epochs=100,verbose=1,validation_split=0.2)
```

```
Epoch 1/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.6040 - loss: 0.6483 - val_accuracy: 0.7969 - val_loss: 0.5162
Epoch 2/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7867 - loss: 0.5208 - val_accuracy: 0.7969 - val_loss: 0.5036
Epoch 3/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8038 - loss: 0.4945 - val_accuracy: 0.7969 - val_loss: 0.5023
Epoch 4/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7861 - loss: 0.5167 - val_accuracy: 0.7969 - val_loss: 0.5013
Epoch 5/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7950 - loss: 0.5049 - val_accuracy: 0.7969 - val_loss: 0.5005
Epoch 6/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7927 - loss: 0.5065 - val_accuracy: 0.7969 - val_loss: 0.4997
Epoch 7/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7974 - loss: 0.5002 - val_accuracy: 0.7969 - val_loss: 0.4990
Epoch 8/100
```

```
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7928 - loss: 0.5049 - val_accuracy: 0.7969 - val_loss: 0.4984
Epoch 9/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8006 - loss: 0.4944 - val_accuracy: 0.7969 - val_loss: 0.4978
Epoch 10/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7926 - loss: 0.5046 - val_accuracy: 0.7969 - val_loss: 0.4974
Epoch 11/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7965 - loss: 0.4986 - val_accuracy: 0.7969 - val_loss: 0.4970
Epoch 12/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8018 - loss: 0.4896 - val_accuracy: 0.7969 - val_loss: 0.4966
Epoch 13/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7980 - loss: 0.4969 - val_accuracy: 0.7969 - val_loss: 0.4963
Epoch 14/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8015 - loss: 0.4911 - val_accuracy: 0.7969 - val_loss: 0.4961
Epoch 15/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7902 - loss: 0.5065 - val_accuracy: 0.7969 - val_loss: 0.4959
Epoch 16/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7964 - loss: 0.4979 - val_accuracy: 0.7969 - val_loss: 0.4957
Epoch 17/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7989 - loss: 0.4956 - val_accuracy: 0.7969 - val_loss: 0.4955
Epoch 18/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7961 - loss: 0.4984 - val_accuracy: 0.7969 - val_loss: 0.4955
Epoch 19/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7952 - loss: 0.5001 - val_accuracy: 0.7969 - val_loss: 0.4953
Epoch 20/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7932 - loss: 0.5035 - val_accuracy: 0.7969 - val_loss: 0.4954
Epoch 21/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7985 - loss: 0.4946 - val_accuracy: 0.7969 - val_loss: 0.4951
Epoch 22/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8030 - loss: 0.4892 - val_accuracy: 0.7969 - val_loss: 0.4955
Epoch 23/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7966 - loss: 0.4968 - val_accuracy: 0.7969 - val_loss: 0.4951
Epoch 24/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7957 - loss: 0.4996 - val_accuracy: 0.7969 - val_loss: 0.4950
Epoch 25/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7966 - loss: 0.4994 - val_accuracy: 0.7969 - val_loss: 0.4949
Epoch 26/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8097 - loss: 0.4818 - val_accuracy: 0.7969 - val_loss: 0.4952
Epoch 27/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7998 - loss: 0.4959 - val_accuracy: 0.7969 - val_loss: 0.4948
Epoch 28/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7886 - loss: 0.5087 - val_accuracy: 0.7969 - val_loss: 0.4951
Epoch 29/100
128/128 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7905 - loss: 0.5071 - val_accuracy: 0.7969 - val_loss: 0.4948
```

```python
y_pred = model.predict(X_test)
```

```
63/63 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
```

```python
y_pred
```

```
array([[0.25373366],
       [0.2521309 ],
       [0.2391832 ],
       ...,
       [0.2391832 ],
       [0.25139228],
       [0.25373366]], dtype=float32)
```

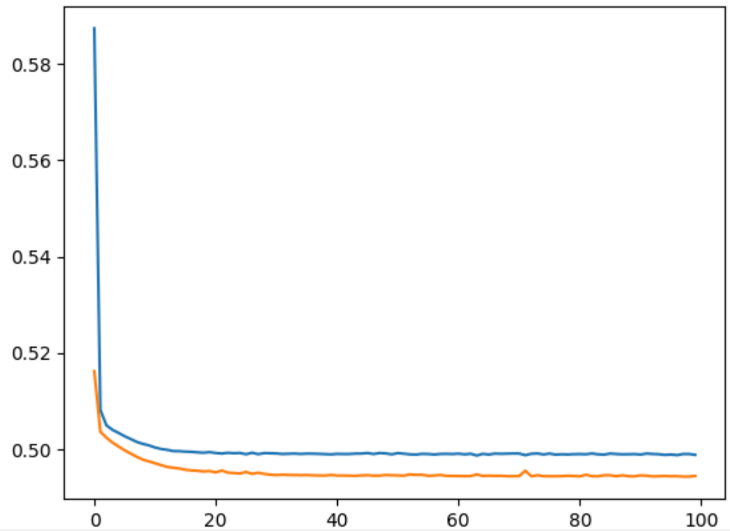
```python
y_pred = y_pred.argmax(axis=-1)
```

```python
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

```
0.7975
```

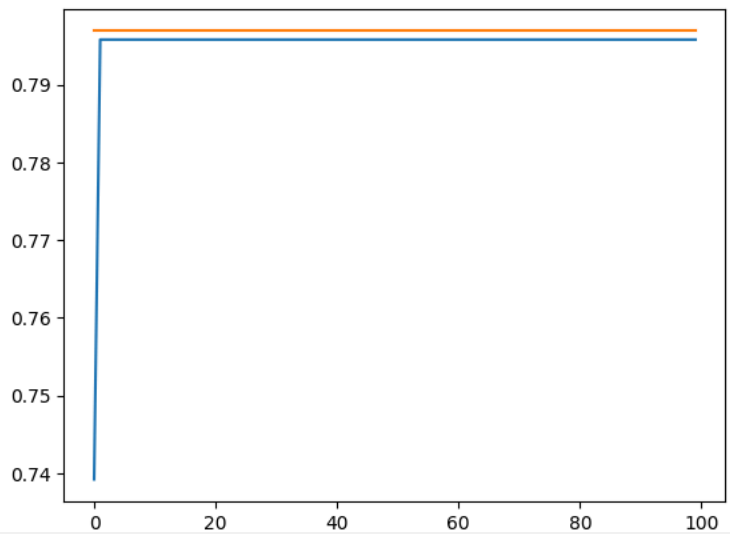
```python
import matplotlib.pyplot as plt

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
```

```
[<matplotlib.lines.Line2D at 0x7f1c45e4f6a0>]
```




```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
```

```
[<matplotlib.lines.Line2D at 0x7f1c45eb6e30>]
```



Start coding or generate with AI.

SCREENSHOTS

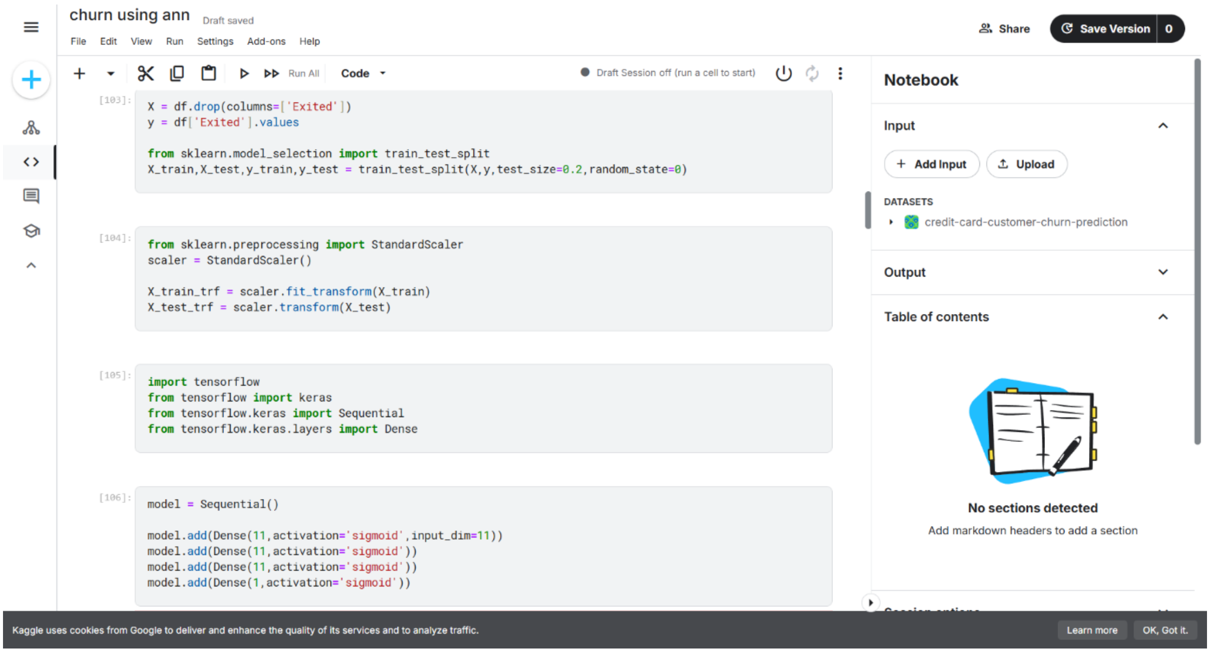


Fig- Code of making ANN model

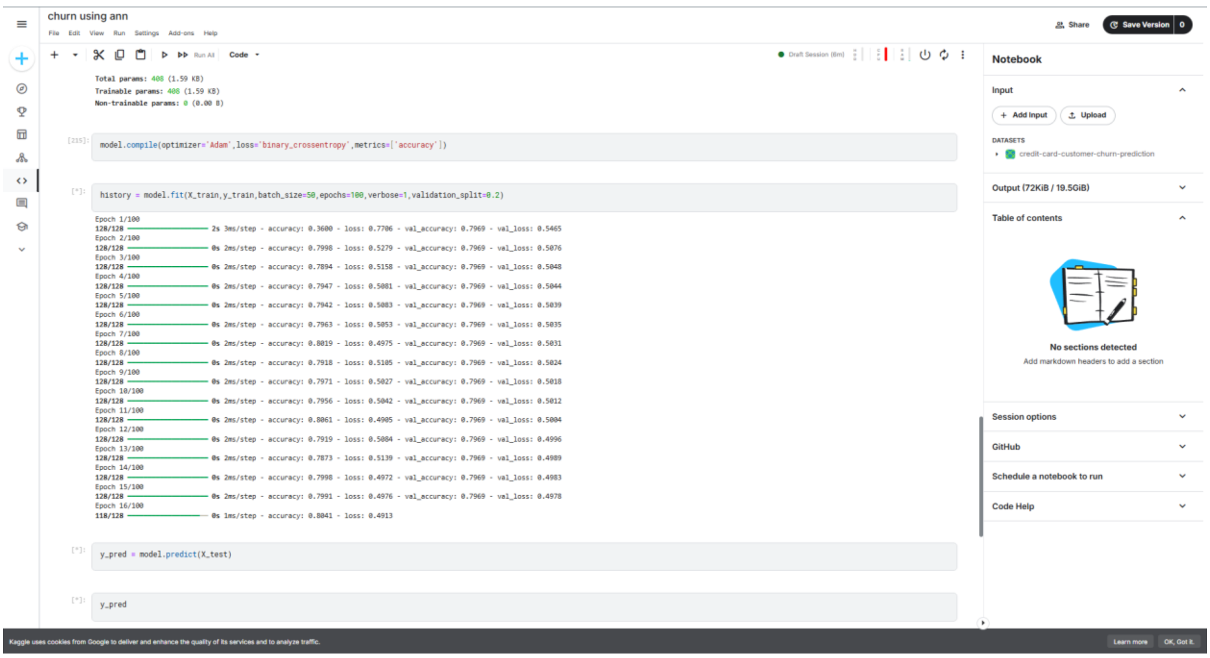


Fig- Training Process

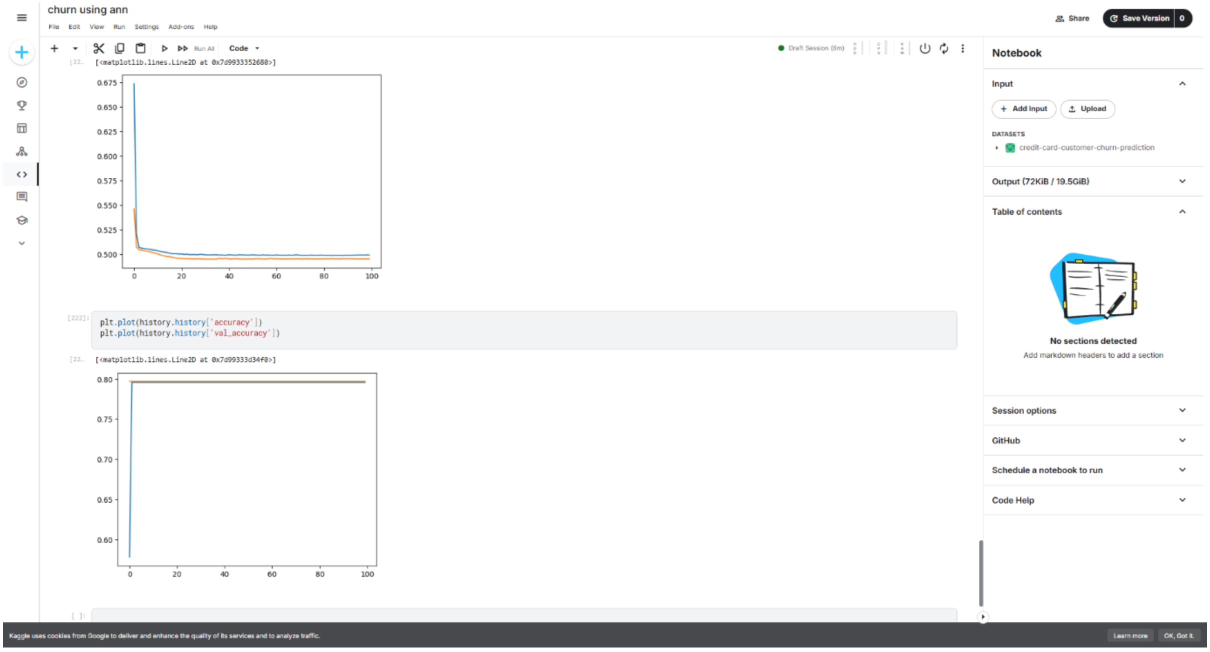


Fig- Representation Using Matplotlib

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

In conclusion, the implementation of Artificial Neural Networks (ANNs) for credit card churn prediction represents a significant advancement in the financial services sector. By leveraging the sophisticated capabilities of ANNs, credit card issuers can accurately identify customers at risk of churning and implement targeted retention strategies that enhance customer loyalty and drive long-term profitability. The ability of ANNs to model complex, non-linear relationships within vast datasets—encompassing customer demographics, transaction histories, payment behaviours, and engagement metrics—enables companies to uncover hidden patterns and trends that traditional methods might miss. This, in turn, allows for more effective and personalized interventions that address the specific needs and preferences of at-risk customers.

The predictive accuracy and adaptability of ANN models are critical assets in the dynamic and competitive financial landscape. ANNs can continuously learn from new data, ensuring that their predictions remain relevant as customer behaviours and market conditions evolve. This adaptability is particularly valuable for credit card issuers, who must navigate shifting consumer preferences and regulatory requirements. Furthermore, the insights generated by ANN-based models can inform strategic decision-making across various aspects of customer relationship management, from marketing and sales to customer service and support.

Future Enhancement

Looking to the future, several enhancements can be made to further improve the efficacy of credit card churn prediction models. Firstly, incorporating additional data sources—such as social media activity, customer feedback, and external economic indicators—can provide a more comprehensive view of customer behaviour and enhance the model’s predictive power. Secondly, the integration of explainable AI techniques can help demystify the decision-making process of ANNs, allowing stakeholders to better understand the factors driving churn predictions and build trust in the model’s outputs. Explainable AI can also aid in regulatory compliance by providing clear and transparent justifications for the model’s decisions.

Moreover, advancements in transfer learning and federated learning offer exciting opportunities for enhancing ANN-based churn prediction models. Transfer learning can enable the model to leverage knowledge gained from related tasks or datasets, reducing the amount of labelled data required for training and improving performance in low-data scenarios. Federated learning, on the other hand, allows for the collaborative training of models across multiple institutions without sharing sensitive customer data, ensuring data privacy and security while benefiting from a larger and more diverse dataset.

Another promising avenue for future research is the exploration of hybrid models that combine ANNs with other machine learning techniques, such as ensemble learning and reinforcement learning. Hybrid models can potentially capture a broader range of patterns and improve overall prediction accuracy by leveraging the strengths of different algorithms. Additionally, ongoing advancements in computational hardware and cloud-based solutions can facilitate the training and deployment of more complex and resource-intensive ANN models, making them accessible to a wider range of organizations.

Summary

In summary, the deployment of Artificial Neural Networks for credit card churn prediction is a transformative approach that empowers credit card issuers to proactively address customer attrition and foster long-term loyalty. As technological advancements continue to unfold, the integration of additional data sources, explainable AI, transfer learning, federated learning, and hybrid models will further enhance the capabilities and effectiveness of ANN-based churn prediction systems. By embracing these future enhancements, credit card companies can stay ahead of the curve, navigate the challenges of customer churn, and unlock new opportunities for growth and innovation in the ever-evolving financial services landscape.