Forecast Commuters Inflow For Airline Industry Using IBM Cloud Services

SMARTBRIDGE EXTERNSHIP- ARTIFICIAL INTELLIGENGE

1. Introduction:

1.1 Overview:

As the coronavirus pandemic wanes,It became very hard to predict the Inflow For Airline Industry In the highly competitive airline industry, customer experience is a major point of differentiation – and digital, mobile-enabled, touchless travel experience is increasingly important during the current COVID-19 pandemic. To increase trust with both passengers and employees, enable an elastic business model and respond better and faster to market changes, major airlines must remove the constraints of the existing legacy architecture, platform, organization, development and operations approaches. Flights landing or departing from international hubs often carry a high proportion of connecting passengers. For example, the vast majority of the flights traveling through Heathrow, the busiest airport in Europe with more than 75 million passengers each year (Heathrow 2016), have at least 25% connecting passengers. Missed connections are the third leading reason for filing a complaint with an airline (MacDonald 2016). Therefore, it is critical that the passengers' transfer journeys are optimized to ensure the airport can fulfil its mission to "give passengers the best airport service in the world".We take a passenger-centric approach to studying transfer passenger flows in the airport. Motivated by our collaboration with Heathrow, we develop a system that predicts the distributions of the connection times of international arriving passengers. With these distributions, airport decision makers can identify passengers with high chance of missing their outbound flights, and take proactive action to reduce missed connections.

passenger flow management have already received some attention in the literature. Wei and Hansen (2006) build an aggregate demand model for air passenger traffic in a hub-and-spoke network, and find that airlines can attract more connecting passengers by increasing service frequency. Barnhart et al. (2014) develop a methodology for modeling delays of the domestic passengers in the United States, and study the key factors that affect the performance of the National Air Transportation System. In their paper, they mention that lack of passenger travel data has made it difficult for researchers to explore passenger centric problems. To the best of our knowledge, no studies relating to passengers' transfer journeys exist that are based on actual data of passengers' travel information.

1.2 Purpose:

The main purpose of this project is to predict the passemgers inflow for the airline in the given date.A predictive system that can be implemented at the airport which helps both passengers and employees and airport managers to understand the key factors that influence passenger's count.

2. Literature Survey:

2.1 Existing problem.

As the coronavirus pandemic wanes,It became very hard to predict the Inflow For Airline Industry In the highly competitive airline industry, customer experience is a major point of differentiation – and digital, mobile-enabled, touchless travel experience is increasingly important during the current COVID-19 pandemic. so it became very necessary to predic the inflow, here we are building
a model using neural networks to predict the inflow.

2.2. Proposed solution.

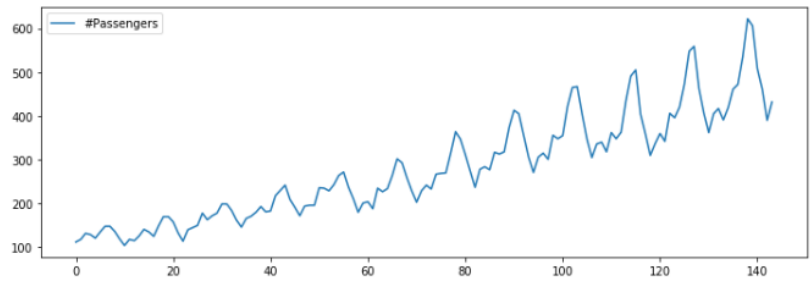
Today, neural networks (NN) are revolutionizing business and everyday life, bringing us to the next level in artificial intelligence (AI). By emulating the way interconnected brain cells function, NN-enabled machines (including the smartphones and computers that we use on a daily basis) are now trained to learn, recognize patterns, and make predictions in a humanoid fashion as well as solve problems in every business sector. Here we are developing a model using the Recurrent neural network (RNN), which predict the Airport Transfer Passenger Flow

3. Theoritical Analysis:

3.1: Block Diagram:

Here We'll be working with the Box and Jenkins (1976) Airline Passengers dataset, which contains time series data on the monthly number of airline passengers between 1949 and 1960. Passenger graph over the years is given below.

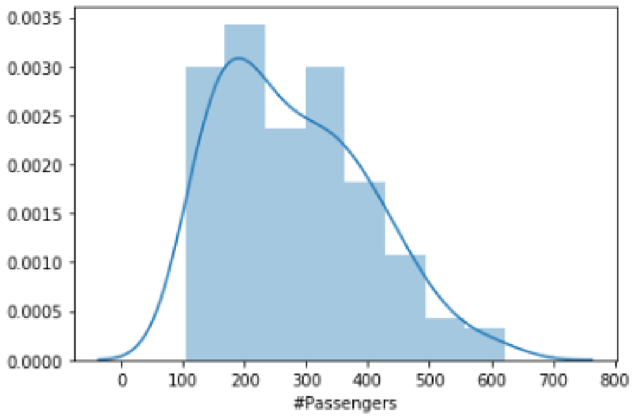
```
In [7]: dataset.plot(figsize=(12,4))
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x245e0a9f358>
```



The bar graph for the passengers over the years is given below.

```
import seaborn as sns
sns.distplot(dataset['#Passengers'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x24698b5f3c8>
```



3.2 Software desingning:

In order to develop this project we need to install the following software/packages:

- Anaconda Navigator
- Jupyter notebook
- sypyder

we have to install some pakages to do this project:

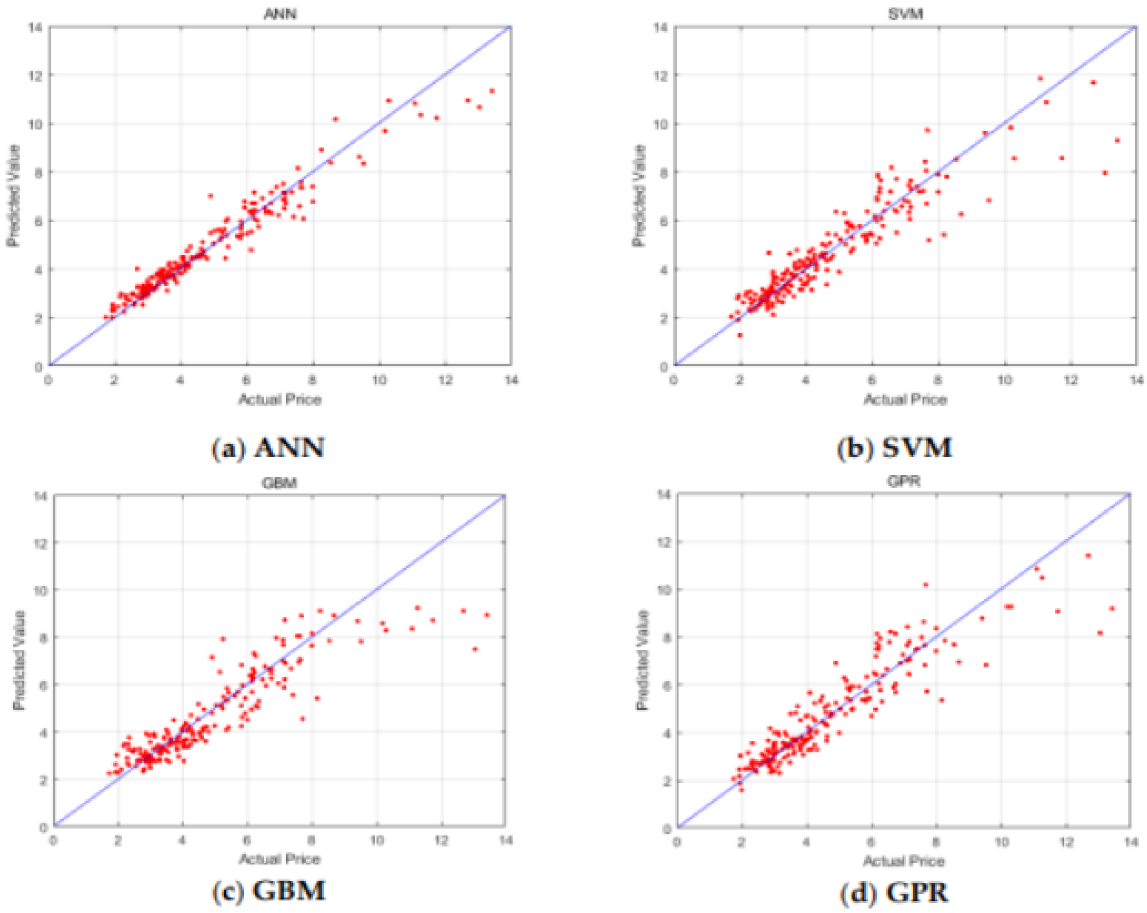
- Tenser flow
- keras
- Flask(which allows you to send HTTP requests using Python.)

4. Experimental Investigations:

A study on the prediction of prices was read and the following observations were made from the research paper.

Based on the before-mentioned machine learning methods, prepared data, forecasting performance evaluation criteria, model validation technique, and selected model parameters, the empirical study is carried out. Observing data of four criteria can easily find that the forecasting performance of ANN and SVM is better than that of GBM and GPR. In particular, ANN is obviously superior to other methods while GBM has the worst behaviour. Overall, the performance ranking is ANN, SVM, GPR, and GBM from strong to weak. intuitively compares the prediction suitcases of four machine learning methods, which contains 214 observed values from January 2001 to October 2018. From the view of the whole tendency, ANN outperforms others, in particular, for the prediction of abnormal values at the beginning of 2009 and the second half of 2010. SVM and GPR are inferior to ANN, but SVM excels the other three methods in terms of price prediction in the middle of 2008. GBM behaves the worst prediction ability among the methods, especially for outliers.

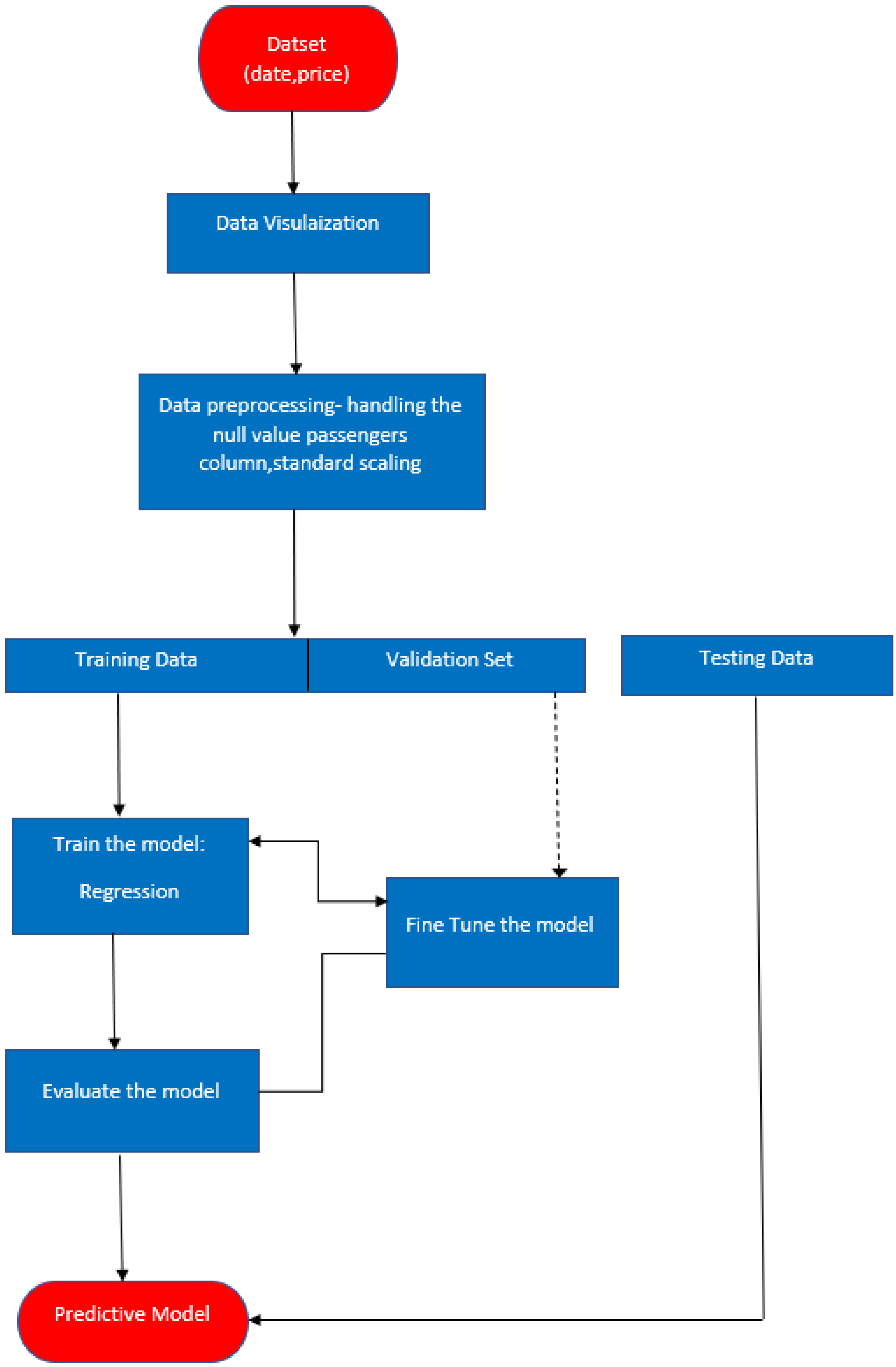
This study motivated me to believe that ANN is one of the most accurate methods of Forecast Commuters Inflow For Airline Industry Using IBM Cloud Services



(a) ANN (b) SVM (c) GBM (d) GPR

Figure 5. Distribution suitcases between predictive values and actual values.

5. Flowchart:



6. Model

6.1 Data Handling:

jupyter air passengers Last Checkpoint: 12 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [153]:
```
#splitting the year,month and day into seperate columns
dataset['Year'] = pd.DatetimeIndex(dataset['Month']).year
dataset['Mon'] = pd.DatetimeIndex(dataset['Month']).month
dataset['Day'] = pd.DatetimeIndex(dataset['Month']).day
```

In [154]:
```
dataset.head()
```

Out[154]:

	Month	#Passengers	Year	Mon	Day
0	1949-01	112	1949	1	1
1	1949-02	118	1949	2	1
2	1949-03	132	1949	3	1
3	1949-04	129	1949	4	1
4	1949-05	121	1949	5	1

In [155]:
```
dataset.drop('Month',axis=1,inplace=True) # deleting the date column as it is unnecessary
```

In [156]:
```
dataset.head()
```

Out[156]:

	#Passengers	Year	Mon	Day
0	112	1949	1	1
1	118	1949	2	1
2	132	1949	3	1
3	129	1949	4	1
4	121	1949	5	1

In [157]:
```
dataset.isnull().any()
```

Out[157]:
```
#Passengers    False
Year           False
Mon            False
Day            False
dtype: bool
```

In [158]:
```
dataset.isnull().sum()
```

Out[158]:
```
#Passengers    0
Year           0
Mon            0
Day            0
dtype: int64
```

6.2 Building the Model:

Creating the model for passenger prediction.

In [169]:
```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

In [170]:
```python
reg=Sequential()
```

In [171]:
```python
reg.add(Dense(units=3,kernel_initializer="random_uniform",activation="relu"))
```

In [172]:
```python
reg.add(Dense(units=100,kernel_initializer="random_uniform",activation="relu"))
```

In [173]:
```python
reg.add(Dense(units=100,kernel_initializer="random_uniform",activation="relu"))
```

In [174]:
```python
reg.add(Dense(units=1,kernel_initializer="random_uniform"))
```

In [175]:
```python
reg.compile(optimizer="rmsprop",loss="mse",metrics=['mse'])
```

In [176]:
```python
history= reg.fit(x_train,y_train,batch_size=200,epochs=4000,validation_data=(x_test,y_test))
```

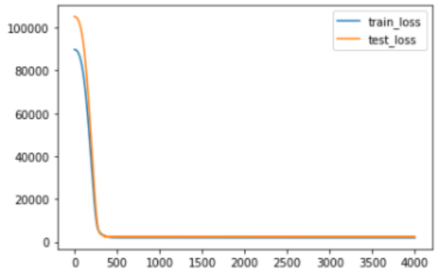
```
Epoch 1/4000
1/1 [==============================] - 0s 469ms/step - loss: 89755.6406 - mse: 89755.6406 - val_loss: 105162.2734 - val_mse:
105162.2734
Epoch 2/4000
1/1 [==============================] - 0s 30ms/step - loss: 89751.2188 - mse: 89751.2188 - val_loss: 105156.8359 - val_mse:
105156.8359
Epoch 3/4000
1/1 [==============================] - 0s 24ms/step - loss: 89746.0859 - mse: 89746.0859 - val_loss: 105150.8203 - val_mse:
105150.8203
Epoch 4/4000
1/1 [==============================] - 0s 26ms/step - loss: 89740.4375 - mse: 89740.4375 - val_loss: 105144.0703 - val_mse:
105144.0703
Epoch 5/4000
1/1 [==============================] - 0s 26ms/step - loss: 89734.0859 - mse: 89734.0859 - val_loss: 105136.4453 - val_mse:
105136.4453
Epoch 6/4000
1/1 [==============================] - 0s 24ms/step - loss: 89726.8984 - mse: 89726.8984 - val_loss: 105127.8281 - val_mse:
105127.8281
Epoch 7/4000
```

In [177]:
```python
y_pred=reg.predict(x_test)
```

```
1/1 [==============================] - 0s 23ms/step
```

6.3 Verifing the model:

In [179]:
```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'],label="train_loss")
plt.plot(history.history['val_loss'],label='test_loss')
plt.legend()
plt.show()
```



Verifying the model

In [180]:
```
sc.transform([[2008,7,30]])
```
Out[180]:
```
array([[15.59686043,  0.1216799 , 29.        ]])
```

In [181]:
```
yp=reg.predict(sc.transform([[2008,7,30]]))
```
```
1/1 [==============================] - 0s 15ms/step
```

In [182]:
```
yp
```
Out[182]:
```
array([[1899.9757]], dtype=float32)
```

7. Results:

The final results are:

In [186]:
```
import joblib
joblib.dump(sc,"scaler")
```
Out[186]:
```
['scaler']
```

In [187]:
```
new = joblib.load('scaler')
```

In [188]:
```
y = new.transform([[2008,7,30]])
```

In [189]:
```
reg.predict(y)
```
```
1/1 [==============================] - 0s 13ms/step
```
Out[189]:
```
array([[1899.9757]], dtype=float32)
```

The final output of flask app is:

8. Applications:

As the industry adapts, passenger airlines and air transportation providers need to look at how they can alleviate health concerns without diminishing the customer experience. In the highly competitive airline industry, customer experience is a major point of differentiation – and digital, mobile-enabled, touchless travel experience is increasingly important during the current COVID-19 pandemic. To increase trust with both passengers and employees, enable an elastic business model and respond better and faster to market changes, major airlines must remove the constraints of the existing legacy architecture, platform, organization, development and operations approaches.

9. Conclusion:

Accuracy of the model- 87.22% The model was successfully deployed by all the team members with the valuable guidance of the mentor under

SmartBridge. Models such as ANN to determine which approach, those based on economic models or on the market, is more accurate and unbiased

10. Future Scope:

- For future work, the effect of emerging machine learning algorithms can be evaluated, such as deep learning and reinforcement learning, on Forecast Commuters Inflow For Airline Industry.
- The next step could be to determine how these results can be helpful to companies to increase trust with both passengers and employees.

11.Bibilography:

- RAKANNIMER , Air Passenger prediction 2020 (https://www.kaggle.com/code/rakannimer/air-passenger-prediction/notebook)
- Abrishami, H.; Varahrami, V. Different methods for forecasting. Cuad. Econ. 2011, 34, 137–144. [CrossRef] Energies 2019, 12, 1680 15 of 17
- Azadeh, A.; Sheikhalishahi, M.; Shahmiri, S. A Hybrid Neuro-Fuzzy Approach for Improvement of Forecasting ..