Model Development Phase -

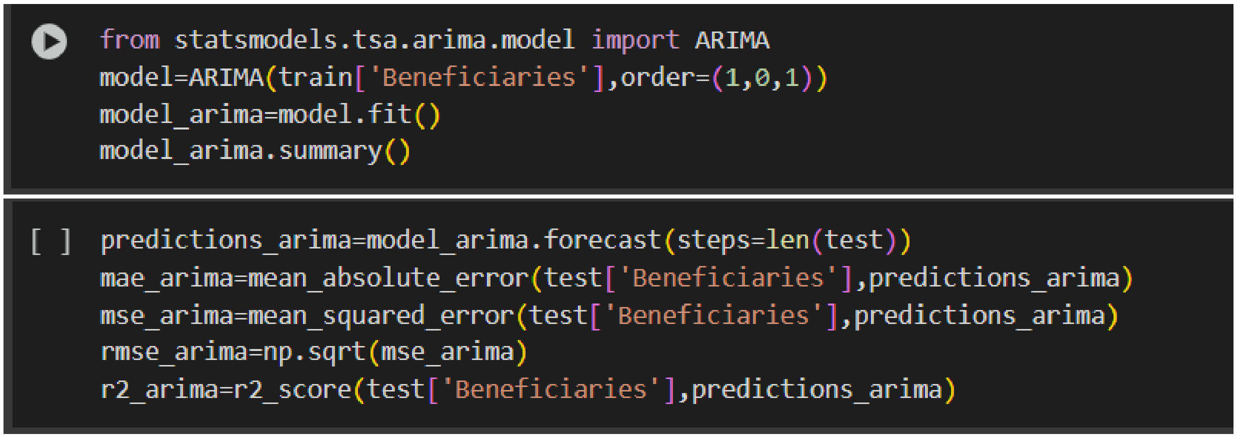
Date	21 JUNE 2025
Team ID	SWTID1749896042
Project Title	Unemployed Insurance Beneficiary Forecasting
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

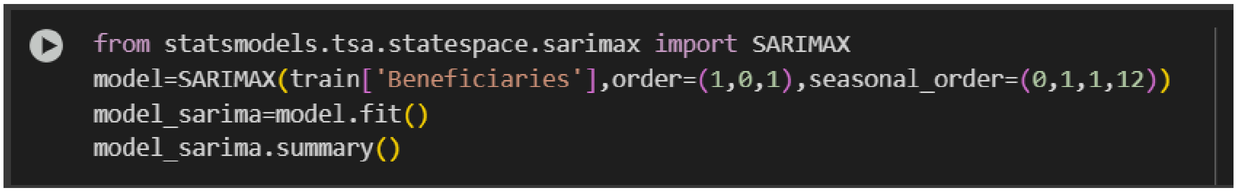
#importing and building the ARIMA model



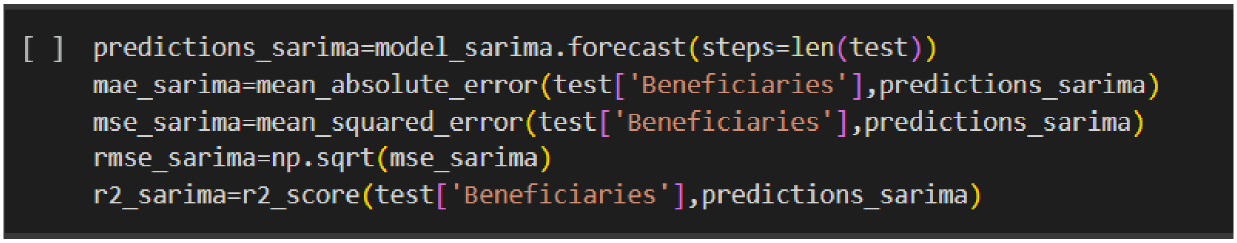
```
from statsmodels.tsa.arima.model import ARIMA
model=ARIMA(train['Beneficiaries'],order=(1,0,1))
model_arima=model.fit()
model_arima.summary()
```

```
[ ]  predictions_arima=model_arima.forecast(steps=len(test))
     mae_arima=mean_absolute_error(test['Beneficiaries'],predictions_arima)
     mse_arima=mean_squared_error(test['Beneficiaries'],predictions_arima)
     rmse_arima=np.sqrt(mse_arima)
     r2_arima=r2_score(test['Beneficiaries'],predictions_arima)
```

#importing and building the SARIMA model

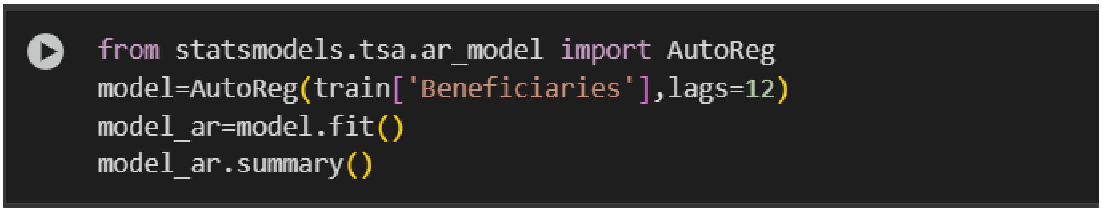


```
from statsmodels.tsa.statespace.sarimax import SARIMAX
model=SARIMAX(train['Beneficiaries'],order=(1,0,1),seasonal_order=(0,1,1,12))
model_sarima=model.fit()
model_sarima.summary()
```

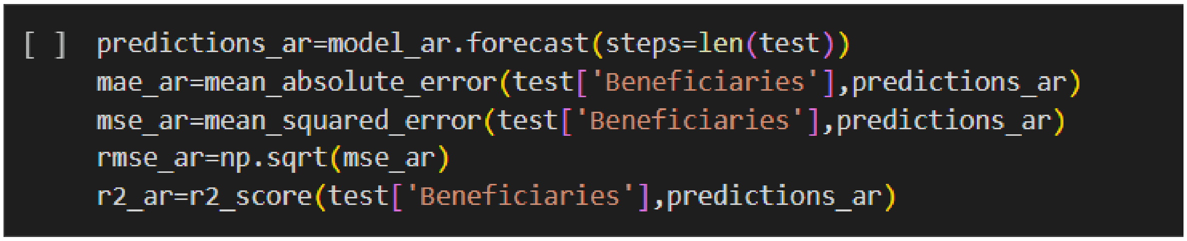


```
predictions_sarima=model_sarima.forecast(steps=len(test))
mae_sarima=mean_absolute_error(test['Beneficiaries'],predictions_sarima)
mse_sarima=mean_squared_error(test['Beneficiaries'],predictions_sarima)
rmse_sarima=np.sqrt(mse_sarima)
r2_sarima=r2_score(test['Beneficiaries'],predictions_sarima)
```

#importing and building the AutoReg model

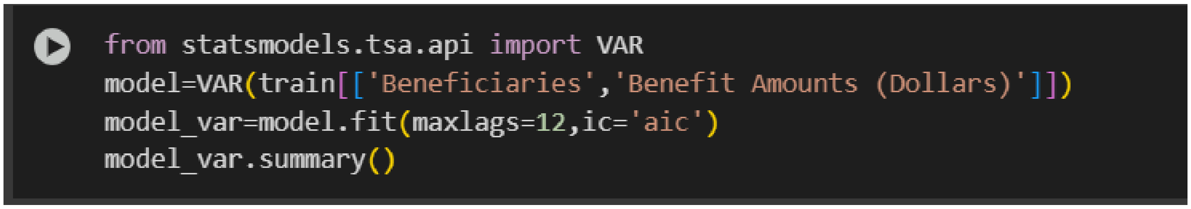


```
from statsmodels.tsa.ar_model import AutoReg
model=AutoReg(train['Beneficiaries'],lags=12)
model_ar=model.fit()
model_ar.summary()
```

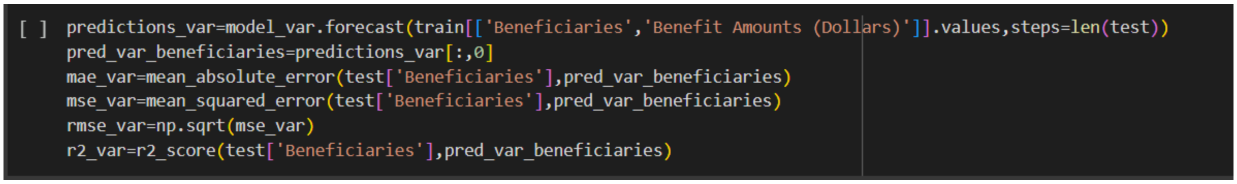


```
predictions_ar=model_ar.forecast(steps=len(test))
mae_ar=mean_absolute_error(test['Beneficiaries'],predictions_ar)
mse_ar=mean_squared_error(test['Beneficiaries'],predictions_ar)
rmse_ar=np.sqrt(mse_ar)
r2_ar=r2_score(test['Beneficiaries'],predictions_ar)
```

#importing and building the VAR (Vector AutoRegression) model

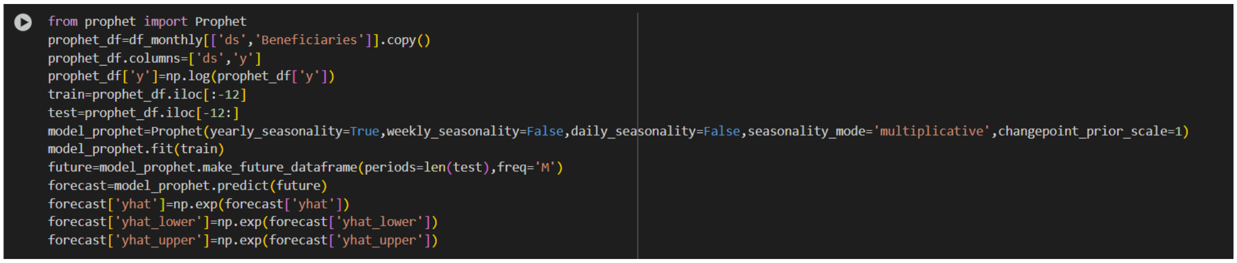


```
from statsmodels.tsa.api import VAR
model=VAR(train[['Beneficiaries','Benefit Amounts (Dollars)']])
model_var=model.fit(maxlags=12,ic='aic')
model_var.summary()
```

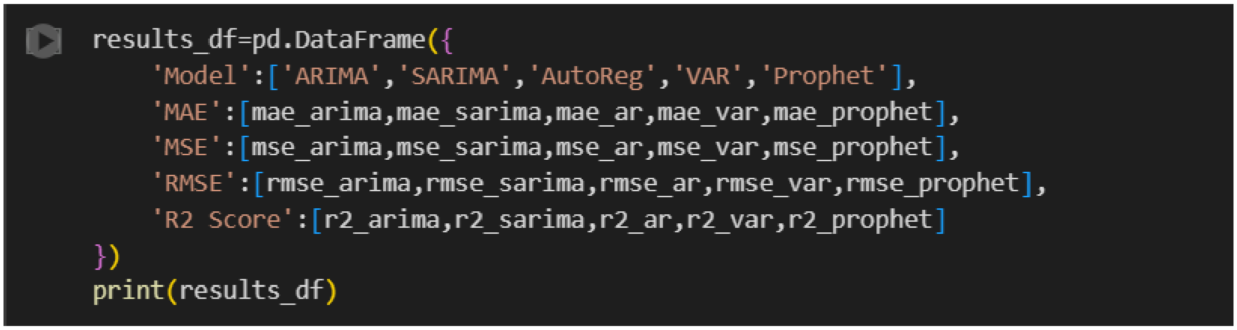


```
[ ] predictions_var=model_var.forecast(train[['Beneficiaries','Benefit Amounts (Dollars)']].values,steps=len(test))
    pred_var_beneficiaries=predictions_var[:,0]
    mae_var=mean_absolute_error(test['Beneficiaries'],pred_var_beneficiaries)
    mse_var=mean_squared_error(test['Beneficiaries'],pred_var_beneficiaries)
    rmse_var=np.sqrt(mse_var)
    r2_var=r2_score(test['Beneficiaries'],pred_var_beneficiaries)
```

#importing and building the Prophet model



```
from prophet import Prophet
prophet_df=df_monthly[['ds','Beneficiaries']].copy()
prophet_df.columns=['ds','y']
prophet_df['y']=np.log(prophet_df['y'])
train=prophet_df.iloc[:-12]
test=prophet_df.iloc[-12:]
model_prophet=Prophet(yearly_seasonality=True,weekly_seasonality=False,daily_seasonality=False,seasonality_mode='multiplicative',changepoint_prior_scale=1)
model_prophet.fit(train)
future=model_prophet.make_future_dataframe(periods=len(test),freq='M')
forecast=model_prophet.predict(future)
forecast['yhat']=np.exp(forecast['yhat'])
forecast['yhat_lower']=np.exp(forecast['yhat_lower'])
forecast['yhat_upper']=np.exp(forecast['yhat_upper'])
```



```
results_df=pd.DataFrame({
    'Model':['ARIMA','SARIMA','AutoReg','VAR','Prophet'],
    'MAE':[mae_arima,mae_sarima,mae_ar,mae_var,mae_prophet],
    'MSE':[mse_arima,mse_sarima,mse_ar,mse_var,mse_prophet],
    'RMSE':[rmse_arima,rmse_sarima,rmse_ar,rmse_var,rmse_prophet],
    'R2 Score':[r2_arima,r2_sarima,r2_ar,r2_var,r2_prophet]
})
print(results_df)
```

Model Validation and Evaluation Report:

NOTE - "Since the objective of this project was to forecast the number of unemployment insurance beneficiaries over time, it focused on time series regression rather than classification. Therefore, classification metrics like precision, recall, and confusion matrix were not applicable."

Model	Classification Report	F1 Score	Confusion Matrix
ARIMA	NA	-	NA
SARIMA	NA	-	NA
AutoReg	NA	-	NA
VAR	NA	-	NA
PROPHET	NA	-	NA