

Team ID: PNT2022TMID23576

PROJECT NAME: DemandEst - AI powered Food Demand Forecaster

Team Leader

The image displays three separate screenshots of a Google Colab notebook titled "Data_preprocessing_and_Model_building (1).ipynb". Each screenshot shows a code cell being run and its corresponding output.

Screenshot 1 (XGBoost):

```
[ ] XG = XGBRegressor()
XG.fit(X_train, y_train)
y_pred = XG.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

[05:36:29] WARNING: /workspaces/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
RMSE: 100.175198689628
```

Screenshot 2 (Linear Regression):

```
[ ] LR = LinearRegression()
LR.fit(X_train, y_train)
y_pred = LR.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSE: 129.7404397040951
```

Screenshot 3 (Lasso):

```
[ ] L = Lasso()
L.fit(X_train, y_train)
y_pred = L.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSE: 129.22746395054898
```

Screenshot 4 (ElasticNet):

```
[ ] EN =ElasticNet()
EN.fit(X_train, y_train)
y_pred= EN.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSE: 131.1336557664208
```

Screenshot 5 (DecisionTree):

```
[ ] DT = DecisionTreeRegressor()
DT.fit(X_train, y_train)
y_pred = DT.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSE: 62.59878547492348
```

Screenshot 6 (KNN):

```
[ ] KNN=KNeighborsRegressor()
KNN.fit(X_train, y_train)
y_pred = KNN.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSE: 66.46696666824401
```

Screenshot 7 (Gradient Boosting):

```
[ ] GB = GradientBoostingRegressor()
GB.fit(X_train, y_train)
y_pred = GB.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSE: 100.16911494445236
```

Team Member 1

colab.research.google.com/drive/1mhHECd4iq-1OdkchMYidY_hDhPD0lx#scrollTo=n7SGIXYO_Vi

Data_preprocessing_and_Model_building (1).ipynb

```
XG = XGBRegressor()
XG.fit(X_train, y_train)
y_pred = XG.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

[05:36:29] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
RMSLE: 100.175198689628
```

```
[ ] LR = LinearRegression()
LR.fit(X_train, y_train)
y_pred = LR.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 129.7404397040951
```

```
[ ] L = Lasso()
L.fit(X_train, y_train)
y_pred = L.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 129.22746395054898
```

Team Member 2

The image displays three separate screenshots of a Google Colab notebook titled "Data_preprocessing_and_Model_building (1).ipynb". Each screenshot shows a code cell being run and its corresponding output.

Screenshot 1 (Top):

```
XG = XGBRegressor()
XG.fit(X_train, y_train)
y_pred = XG.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

[05:36:29] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
RMSLE: 100.175198689628
```

Screenshot 2 (Middle):

```
LR = LinearRegression()
LR.fit(X_train, y_train)
y_pred = LR.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 129.7404397040951
```

Screenshot 3 (Bottom):

```
EN = ElasticNet()
EN.fit(X_train, y_train)
y_pred= EN.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 131.1336557664208
```

```
DT = DecisionTreeRegressor()
DT.fit(X_train, y_train)
y_pred = DT.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 62.59878547492348
```

```
KNN=KNeighborsRegressor()
KNN.fit(X_train, y_train)
y_pred = KNN.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 66.4669666824401
```

Screenshot 4 (Bottom-most):

```
GB = GradientBoostingRegressor()
GB.fit(X_train, y_train)
y_pred = GB.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 100.16911494445236
```

Team Member 3

The image displays three separate screenshots of a Google Colab notebook titled "Data_preprocessing_and_Model_building (1).ipynb". Each screenshot shows a code cell being run and its corresponding output.

Screenshot 1 (Top):

```
XG = XGBRegressor()
XG.fit(X_train, y_train)
y_pred = XG.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

[05:36:29] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
RMSLE: 100.175198689628
```

Screenshot 2 (Middle):

```
LR = LinearRegression()
LR.fit(X_train, y_train)
y_pred = LR.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 129.7404397040951
```

Screenshot 3 (Bottom):

```
EN = ElasticNet()
EN.fit(X_train, y_train)
y_pred= EN.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 131.1336557664208
```

```
DT = DecisionTreeRegressor()
DT.fit(X_train, y_train)
y_pred = DT.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 62.59878547492348
```

```
KNN=KNeighborsRegressor()
KNN.fit(X_train, y_train)
y_pred = KNN.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 66.4669666824401
```

Screenshot 4 (Bottom-most):

```
GB = GradientBoostingRegressor()
GB.fit(X_train, y_train)
y_pred = GB.predict(X_val)
y_pred[y_pred<0] = 0
from sklearn import metrics
print('RMSLE:',100*np.sqrt(metrics.mean_squared_log_error(y_val, y_pred)))

RMSLE: 100.16911494445236
```