To,

IITD-AIA Foundation on Smart Manufacturing

Subject: Weekly Progress Report for Week-6th

Respected sir,

Following is the required progress report to the best of my knowledge considering relevant topics to be covered:

1. *Linear Regression implimentastion .*

2. *Logistic Regression for classification .*

3. Support Vector Machines.

4. Decision tree classification.

5. Ensemble Learning.

6. Model Selection Techniques.

## Day wise work done explanation of the week.

Day 11, July

***Explored on Linear Regression implimentastion***

*Code:*

*import numpy as np*

*import pandas as pd import os*

*from sklearn.linear_model import LinearRegression*

*from sklearn.preprocessing import OneHotEncoder, StandardScaler*

*import warnings*

```
import matplotlib.pyplot as plt

%matplotlib inline

for dirname,filenames in os.walk('/kaggle/input'):

for filename in filenames:

print (os.path.join(dirname, filename))

data = pd.read_csv("C:\Users\Nandan\Desktop\fsm\AC-Piston-Checks-main\data\de

data.head()

data.shape

print(data["image_path'].unique())

print(data['target'].unique())

##check missing or null values
```

```
data.isnull().sum()

data.describe() data.head()

import seaborn as sns

sns.pairplot(data)

numeric_data = data.select_dtypes (include=[np.number])

# Calculate the correlation matrix

corrmat = numeric_data.corr()

top_corr_features = corrmat. index

plt.figure(figsize=(5, 5))

# Plot heat map

g= sns.heatmap(numeric_data[top_corr_features].corr(), annot=True, cmap="RdY1Gn")
plt.show()
```

Reference:

https://www.kaggle.com/s/1517530

https://www.youtube.com/watch?v=nOYW31rrkig&pp=ygUnaW5kZXBlbmRlbnQgYW5kIGRlcGVuZGVudCBmZWF0dXJlIGluIG1s

Day 12 , July

*Logistic Regression for classification*

*Steps:*

*It predicts categories based on given features.*

*The algorithm learns from data to calculate probabilities of category membership.*

*It uses a sigmoid function to model the relationship between features and probabilities.*

*During training, the algorithm adjusts parameters to minimize prediction errors.*

*The trained model can be used to classify new instances by applying a threshold to probabilities.*

*Evaluation metrics like accuracy, precision, and recall can assess the model's performance.*

*Logistic regression can handle binary classification and can be extended to multi-class problems*

Reference:

https://www.kaggle.com/code/faressayah/logistic-regression-for-binary-classification-task

https://www.youtube.com/watch?

v=zM4VZR0px8E&pp=ygUmTG9naXN0aWMgUmVncmVzc2lvbiBmb3IgY2xhc3NpZmljYXRpb24%3D

Day 13, July

Support Vector Machines

Steps:

1. Preprocess the Data.

2. Split the Data into training and testing sets.

3. Choose the SVM type (classification or regression).

4. Select a kernel function (linear, polynomial, RBF, sigmoid, etc.).

5. Train the SVM model.

6. Tune the hyperparameters.

7. Evaluate the model's performance.

8. Make predictions on new data.

Reference:

https://www.kaggle.com/discussions/getting-started/124508

https://www.youtube.com/watch?v=5pZ-_MSM0rU&pp=ygUwIFN1cHBvcnQgIFN1cHBvcnQgVmVjdG9yIE1hY2hpbmVzVmVjdG9yIE1hY2hpbmVz

Day 14 ,july

Decision tree classification .

It splits the data based on feature values and assigns class labels to the resulting leaf nodes.

Steps:

1 Data preparation

2 Feature selection

3 Tree construction

4 Stopping criteria

5 Pruning (optional)

6 Prediction

7 Evaluation

Example code for decision tree classification

using Python's scikit-learn library:

```
# Import the necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Load the dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create an instance of the DecisionTreeClassifier
clf = DecisionTreeClassifier()

# Train the classifier using the training data
clf.fit(X_train, y_train)

# Make predictions on the testing data
```

```
y_pred = clf.predict(X_test)


# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Reference:

https://www.youtube.com/watch?v=ynTCUngbFHA&pp=ygUcRGVjaXNpb24gdHJlZSBjbGFzc2lmaWNhdGlvbg%3D%3D

https://www.kaggle.com/rishidamarla/decision-tree-classification

*Day 15 , July :*

Ensemble Learning.

Ensemble learning combines multiple models to improve prediction accuracy and robustness. It trains different models, combines their predictions, and produces a more accurate final result

Steps :

1. Train multiple models (base learners).
2. Combine their predictions.
3. Obtain the final ensemble prediction.
4. Achieve improved accuracy and robustness

2) Ensemble Techniques

- 
- 2.1 Max Voting / Voting Classifier
- 
- 2.2 Averaging
- 
- 2.3 Weighted Averaging
- 
- 2.4 Stacking
- 
- 2.5 Blending

- 
- 2.6 Bagging
- 
- 2.7 Boosting

Reference:

https://www.kaggle.com/code/pavansanagapati/ensemble-learning-techniques-tutorial

https://www.youtube.com/watch?v=KIOeZ5cFZ50&pp=ygURRW5zZW1ibGUgTGVhcm5pbmc%3D

Day 16 , July ;

**Model Selection Techniques** .

These techniques help in determining which model will likely generalize well to new, unseen data.

Steps :

Define the problem and evaluation criteria.Preprocess data and engineer features.Select candidate models.Split data into training, validation, and test sets.Train and evaluate models on training and validation sets.Compare model performance using metrics.Select best-performing model.Assess selected model using test set.

Reference:

https://medium.com/international-school-of-ai-data-science/use-these-methods-to-select-features-for-ml-model-1450e068ebdc?source=search_post---------0----------------------------

https://www.kaggle.com/code/apapiu/regularized-linear-models

https://www.youtube.com/watch?v=yN7ypxC7838&pp=ygUbIG1vZGVsIHNlbGVjdGlvbiB0ZWNobmlxdWVz