Options Pricing Group Project

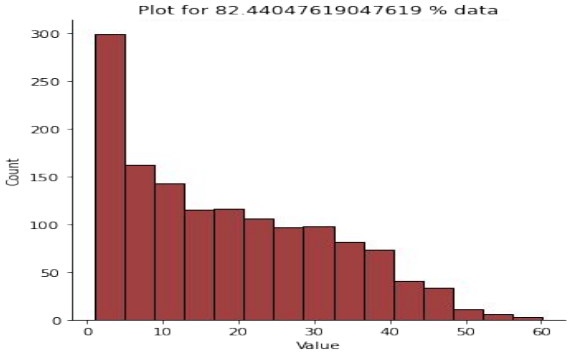
By:
Himani Desai
Mridula Singhal
Pranay Bhaskar Mallipudi
Pratik Khadse
Shailendra Narwaria

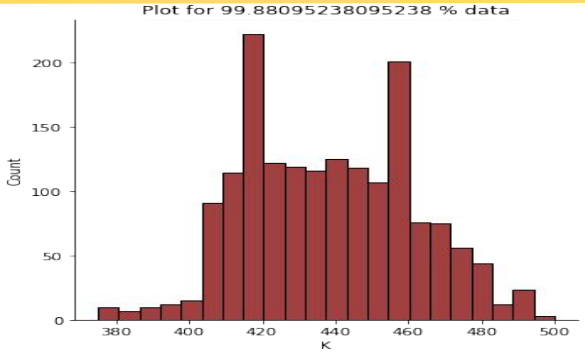
Introduction

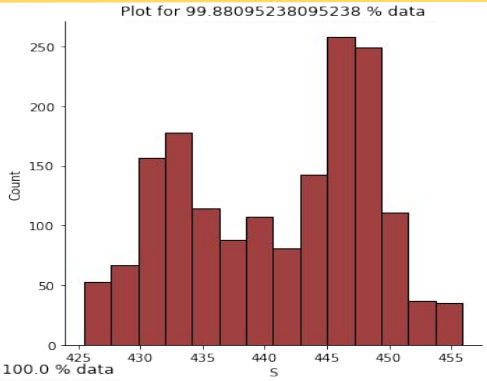
Examined European call option pricing data on the **S&P 500** by **regression and classification techniques** to value an option

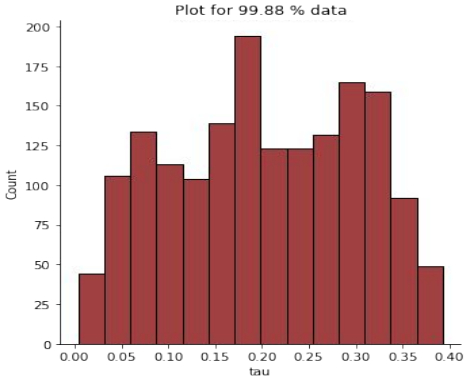
Furthermore, we have also built an **interactive dashboard** which generates the regression and classification results based on user input

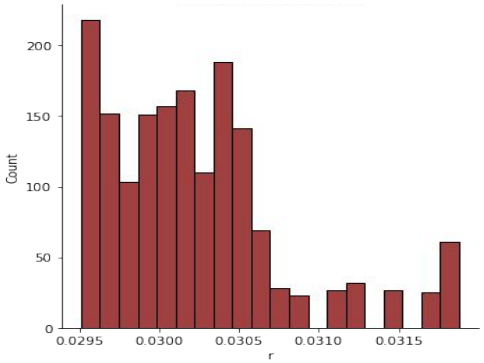
Exploratory Data Analysis





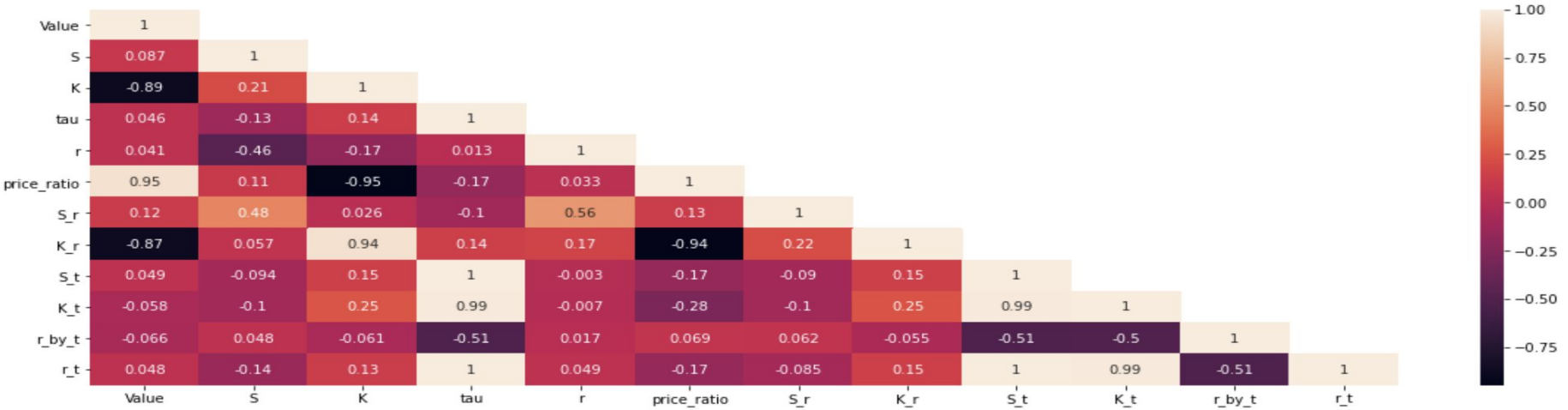




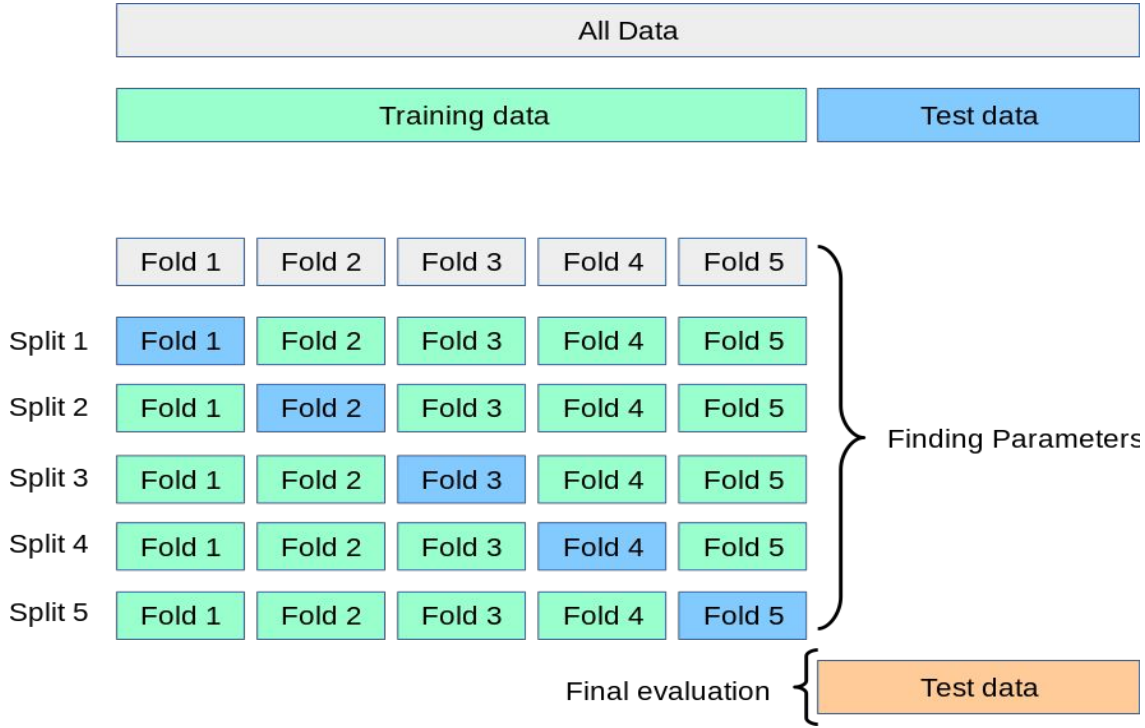


New Variables

```
df["price_ratio"] = df["S"]/df["K"]
df['S_r'] = df['S']*df['r']
df['K_r'] = df['K']*df['r']
df['S_t'] = df['S']*df['tau']
df['K_t'] = df['K'] * df['tau']
df['r_by_t'] = df['r']/df['tau']
df["r_t"] = df["tau"]*df["r"]
```



Cross Validation



- **Regression:** Simple KFold
- **Classification:** Stratified KFold

Modeling Results

	method	cv score	mean cv r2 score	std score	r2 score on test data
0	Linear Regression	[0.95258, 0.93119, 0.93918, 0.93387, 0.93426]	0.938218	0.007632	0.931445
1	Decision tree	[0.99561, 0.99493, 0.99543, 0.9951, 0.99447]	0.995107	0.000398	0.995587
2	Random Forest	[0.99721, 0.99751, 0.99819, 0.99749, 0.99805]	0.997688	0.000370	0.998012
3	KNN Regressor	[0.98902, 0.99005, 0.99063, 0.98078, 0.98937]	0.987969	0.003637	0.992260
4	Neural Network Regression	[0.99556, 0.99457, 0.9945, 0.9936, 0.99555]	0.994758	0.000735	0.995992
5	XGBoost Regressor	[0.99847, 0.99802, 0.99833, 0.9978, 0.99863]	0.998249	0.000299	0.998754
6	Support Vector Machine	[0.95235, 0.98125, 0.97862, 0.97526, 0.9838]	0.974256	0.011311	0.990714

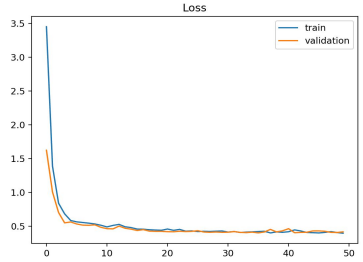
	method	cv error	mean cv error	std cv error	error on test data
0	Logistic Regression	[0.09362, 0.05957, 0.08085, 0.10256, 0.12393]	0.09211	0.02149	0.093439
1	Decision tree	[0.09362, 0.05532, 0.05532, 0.10684, 0.11538]	0.08530	0.02544	0.089463
2	Random Forest	[0.0766, 0.04681, 0.0383, 0.0641, 0.11111]	0.06738	0.02560	0.071571
3	Knn Classifier	[0.08936, 0.05957, 0.05532, 0.05983, 0.08974]	0.07076	0.01542	0.069583
4	Linear Discriminant Analysis	[0.10213, 0.06383, 0.08085, 0.10256, 0.12393]	0.09466	0.02057	0.089463
5	Neural Network	[0.10213, 0.04255, 0.04681, 0.05983, 0.09402]	0.06907	0.02449	0.083499
6	XGBoost Classifier	[0.08085, 0.05532, 0.05106, 0.06838, 0.11111]	0.07334	0.02158	0.073559
7	Support Vector Machine	[0.10213, 0.05106, 0.0383, 0.06838, 0.10256]	0.07249	0.02618	0.081511

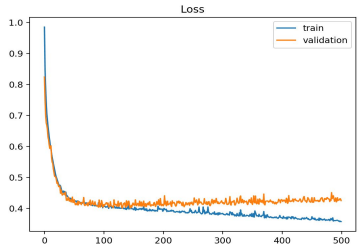
Validate: Learning Curves

Good fit

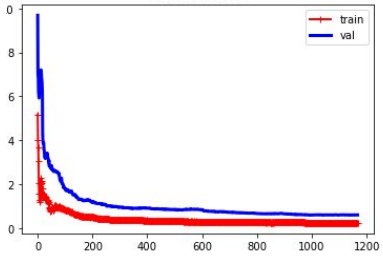
Bad fit

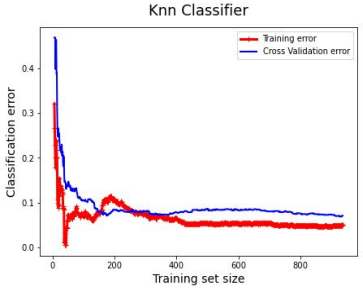
Example

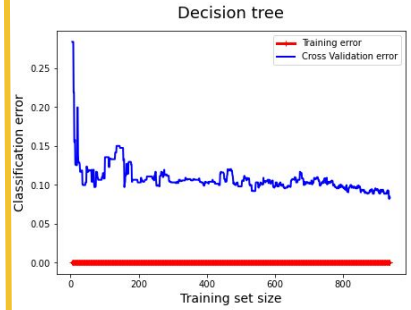


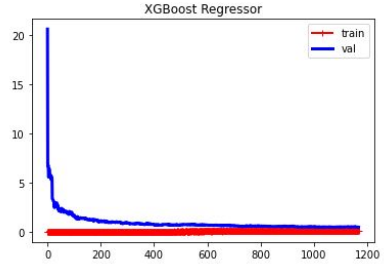


Our Models



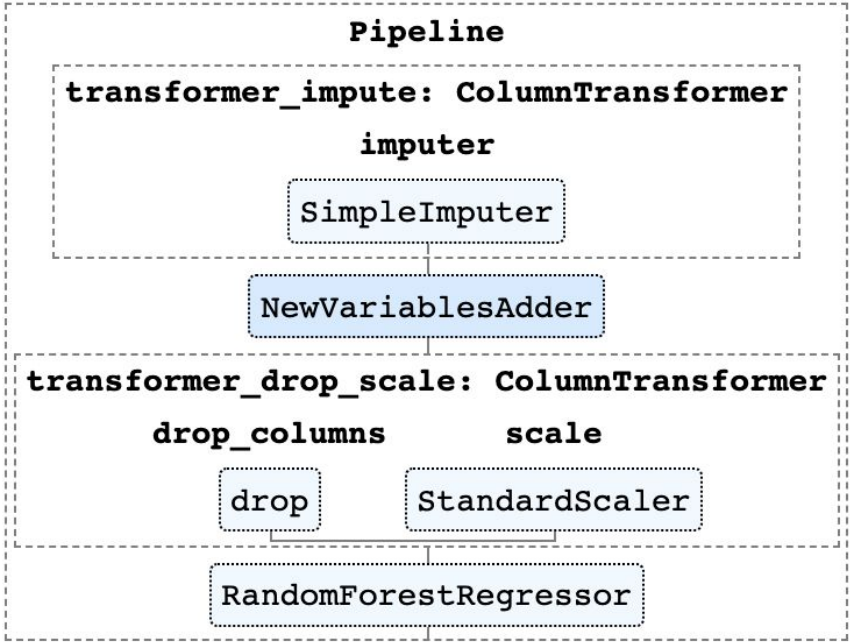


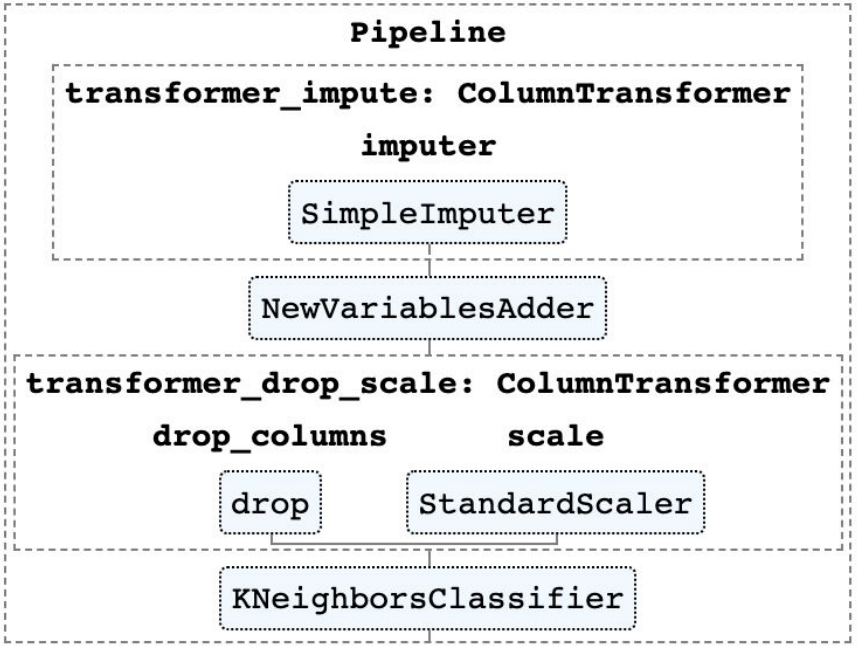




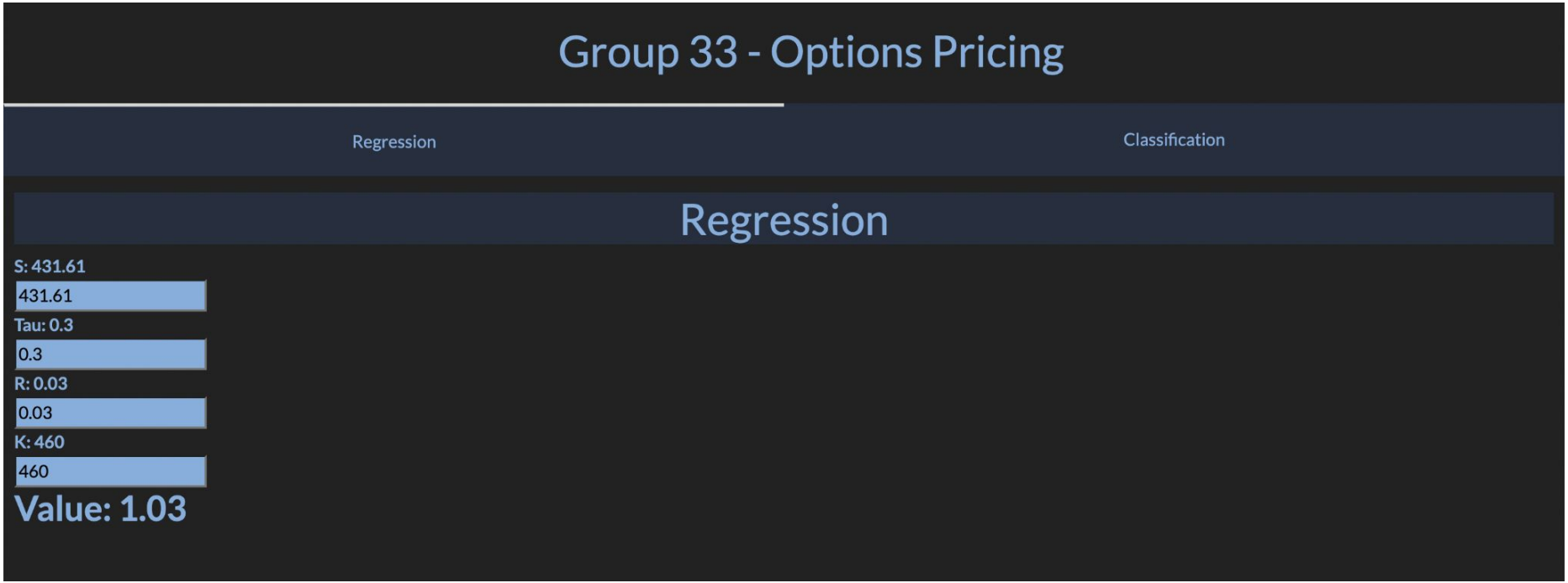
Above & Beyond

Scalability and Production deployment





Interactive UI for Users



Conclusion

Highest Accuracy for regression problem is found in **Random Forest Regressor**

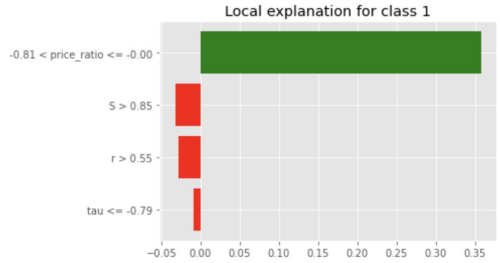
(above the 94% required)

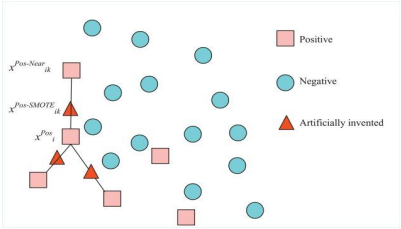
Least error for classification problem is found in **KNN Classifier**

(below the required 8%)

Future Scope

- Building Better Variables Based on Domain Expertise
- F1-scores, Precision and Recall for Classification Model
- SMOTE and LIME





Conclusion

1) Prediction **accuracy** or **interpretation** is more important?
2) Machine learning models may **outperform** Black-Scholes in terms of predicting option values?
3) No **variable selection**?
4) Use your model to predict option values for **Tesla stocks**? Why?

Thank You