Options Pricing Project

Group 34

Scope

European Call Option pricing data is used to build machine learning models to predict option prices in the future. Training data has 1680 options. The Black-Scholes option pricing formula provides an approach for valuing such options.

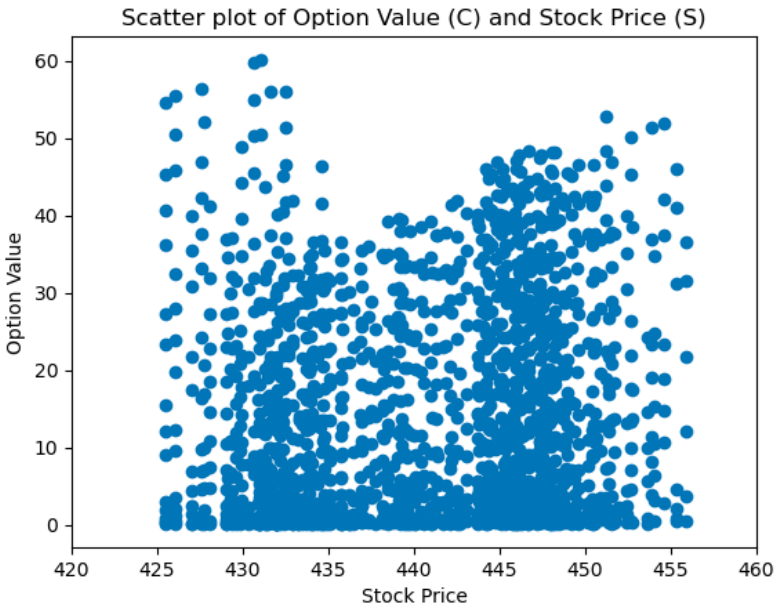
$$C\_pred = S\Phi(d_1) - Ke^{-r\tau}\Phi(d_2)$$

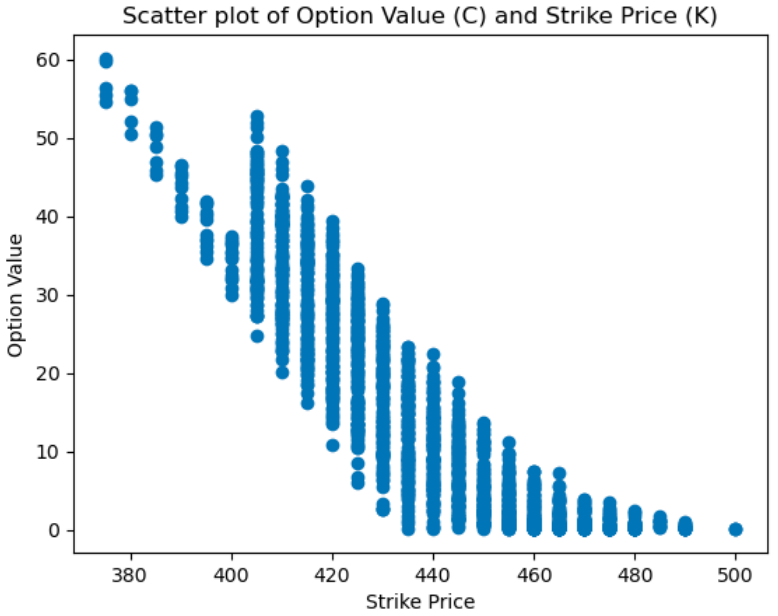
Fields in dataset:
Value(C): current option value
S: current asset value
K: strike price of option
r: annual interest rate
tau: time until expiration of option
BS: Black-Scholes formula that was applied this data to get C.

Process Overview

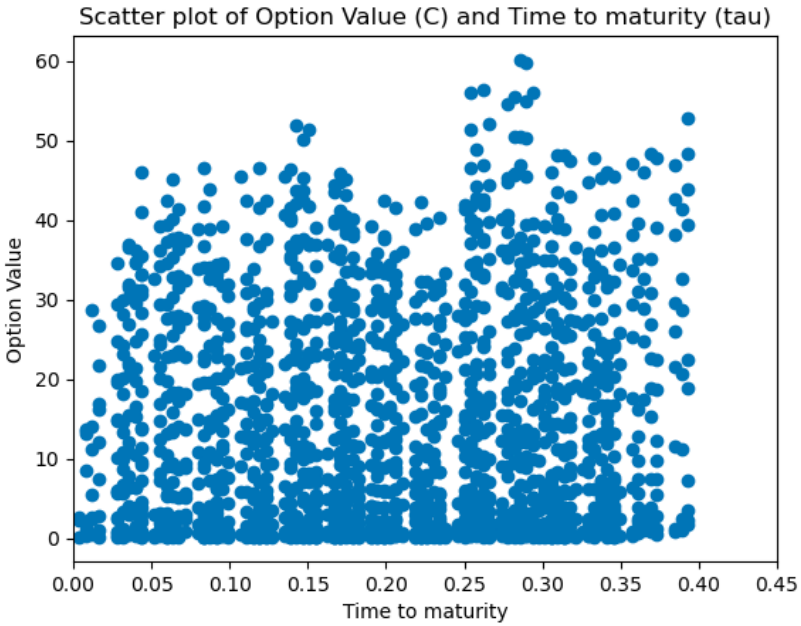
Exploratory Data Analysis	Data Cleaning & Pre-Processing	Modeling	Picking the best model
Computed summary statistics and visualized variables.	Explored data to find missing and erroneous values. Deciding whether or not to scale variables before modeling.	Explored different machine learning models for predicting option value (regression) and whether predicted option value was higher or lower than the Black-Scholes model (classification).	K-fold (10-fold) cross validation is performed to pick the final model based on R^2 and classification error of models.

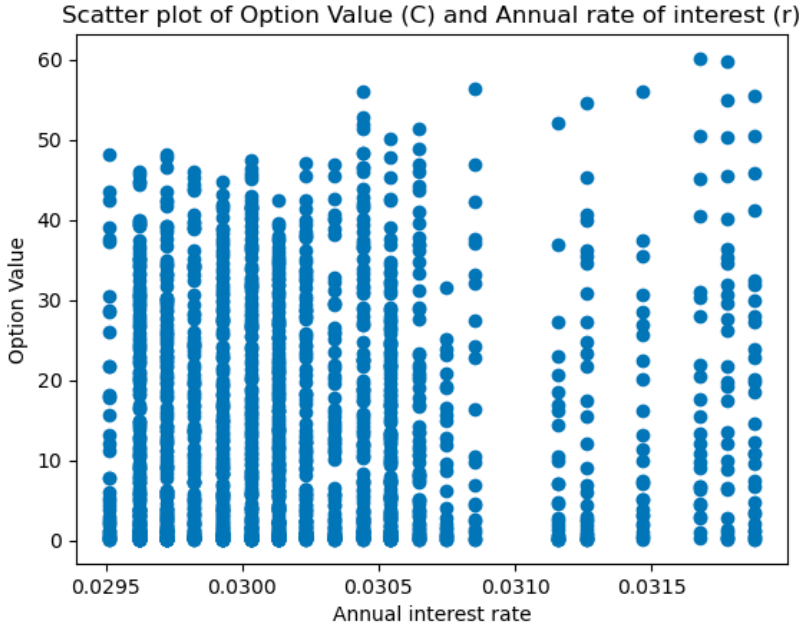
Exploratory Data Analysis





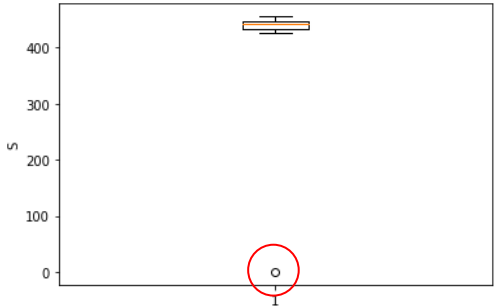
Exploratory Data Analysis

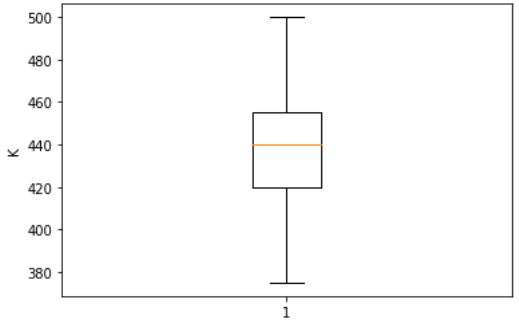


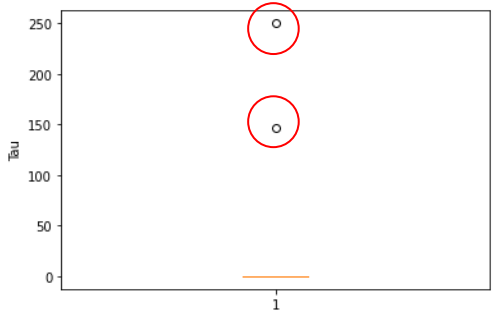


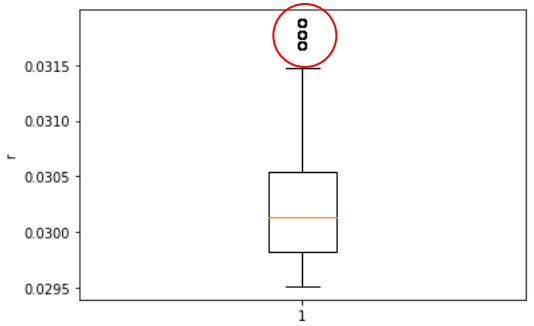
Data Cleaning

- Using Boxplot for each independent variable to identify outliers/data anomalies









Data Cleaning & Standardization

- Removed rows with missing values (removed 2 rows)
- Removed rows with erroneous values for Stock price (S = 0)
- Removed rows with erroneous values for Time to maturity (tau > 2).

Scaling the data:

- Used MinMax/Standard scaler to rescale the data.

Regression

Simple Regression on Values using OLS

OLS Regression Results

Dep. Variable:	Value	**R-squared:**	0.912
Model:	OLS	**Adj. R-squared:**	0.912
Method:	Least Squares	**F-statistic:**	4319.
Date:	Thu, 21 Apr 2022	**Prob (F-statistic):**	0.00
Time:	15:16:40	**Log-Likelihood:**	-4767.7
No. Observations:	1675	**AIC:**	9545.
Df Residuals:	1670	**BIC:**	9573.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-22.0596	11.235	-1.963	0.050	-44.096	-0.023
S	0.6216	0.016	39.738	0.000	0.591	0.652
K	-0.5903	0.005	-130.001	0.000	-0.599	-0.581
tau	31.6699	1.046	30.265	0.000	29.617	33.722
r	509.1620	206.829	2.462	0.014	103.490	914.834

Equation:

Values = 0.6216*S - 0.5903*K + 31.67*tau + 509.16*r

Very low p-values of all independent variables

Regression Approach

1. Define variables
    a. Y = Value
    b. X = S, K, tau, r
2. Standardization
3. Used GridSearch to tune the parameters
4. Cross-validation (KFold)

Regression Methods Tried:

1. Linear Regression
2. KNN
3. Random Forest
4. Decision Trees
5. SVM
6. Lasso & Ridge

Mean R2 Score: 99.92%

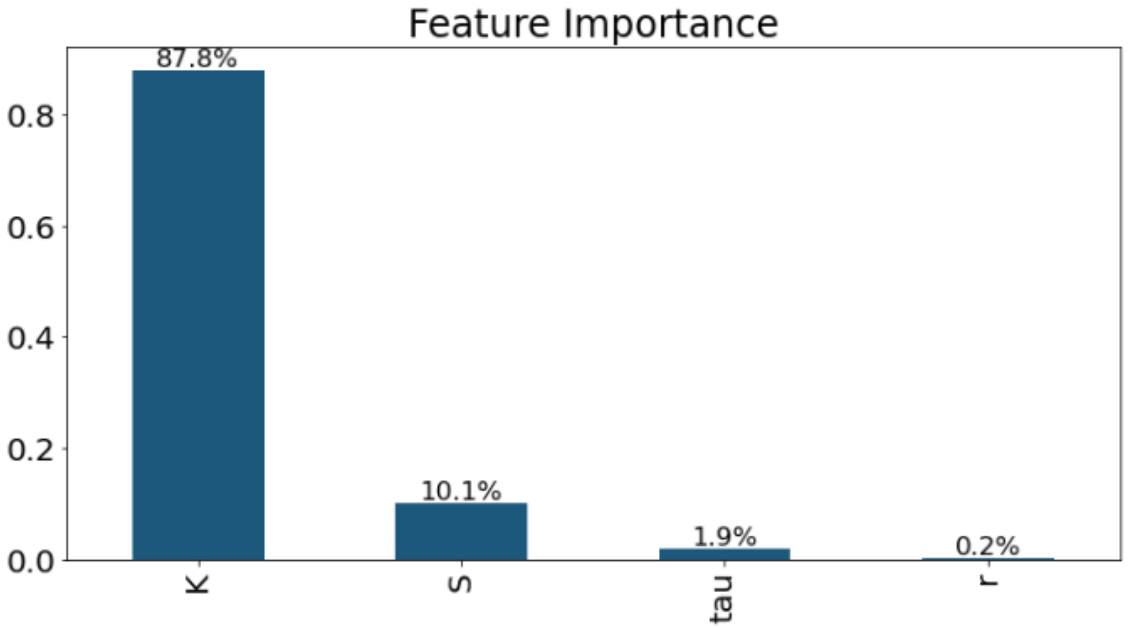
Random Forest Regression

- Random forest is an ensemble learning algorithm based on decision tree learners
- sklearn.RandomForestRegressor
- Used RandomizedSearchCV to tune Hyperparameters

```
Average accuracy: 0.9992997392270304
```

n_estimators = 800
criterion = 'squared_error'
min_samples_leaf = 1
Min_samples_split = 2
max_features = auto
bootstrap=True
max_depth = 100

Random Forest Feature Importance



Mean R2 Score: 99.86%

SVM Regression

- Support Vector Regression is a supervised learning algorithm that uses the same principle as SVM
- sklearn.svm.SVR()
- Used GridSearchCV to tune Hyperparameters

kernel='rbf'
gamma = 1
C = 100
epsilon = 0.2

```
r squared of 10-folds: [0.9988503  0.998309   0.99865447 0.99845497 0.99863093 0.9985857
 0.99846763 0.99891352 0.99831951 0.99859697] (mean r squared: 0.9985783005482588 )
```

Mean R2 Score: 99.28%

Decision Tree

- Decision tree trains a model in the structure of a tree to predict data
- sklearn.tree.DecisionTreeRegressor()
- Used Default Parameters

```
criterion='squared_error'
max_depth=None
min_samples_split=2
min_samples_leaf=1
```

```
r squared of 10-folds: [0.99042643 0.99448682 0.99285298 0.99237916 0.99402658 0.99107275
 0.99477176 0.99395908 0.99417086 0.99153096] (mean r squared: 0.9929677390221461 )
```

Regression Model Comparisons

Method/Metric	R-Squared
Random Forest	**0.9992**
SVM	0.9986
Decision Tree	0.9930

Classification

Classification Approach

1. Define variables
    a. Y = BS (changed to binary (0 or 1))
    b. X = S, K, tau, r
2. Standardization
3. Define the parameters
4. Calculate the accuracy/classification error
5. Cross-validation (KFold)
6. ROC/AUC Curve

Classification Methods Tried:

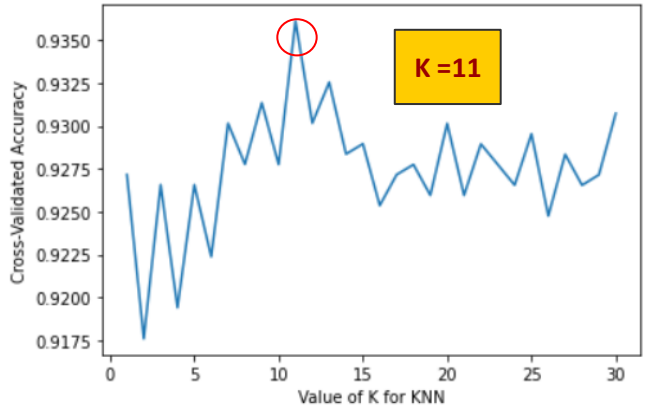
1. Logistic Regression
2. KNN
3. Random Forest
4. Decision Trees
5. SVM

Classification Error: 7.52%

K-Nearest Neighbors (KNN)

- Supervised learning method to classify an object by the number of “k” closest neighbors based on the position of the said object
- Use KNeighborsClassifier to find the optimal K (neighbors) to run the classification

```
mean classification error: 0.07523167949814646
```



Classification Error: 6.57%

SVM Classification

- Supervised learning method to separate two or more different classes through the use of hyperplane and support vectors
- Utilizes kernel to build the boundary, bearings/margins
- Use GridSearchCV()

```
mean classification error: 0.06566153407470754
```

Kernel='rbf'
gamma=100
C=1
Epsilon=0.2

Classification Error: 5.07%

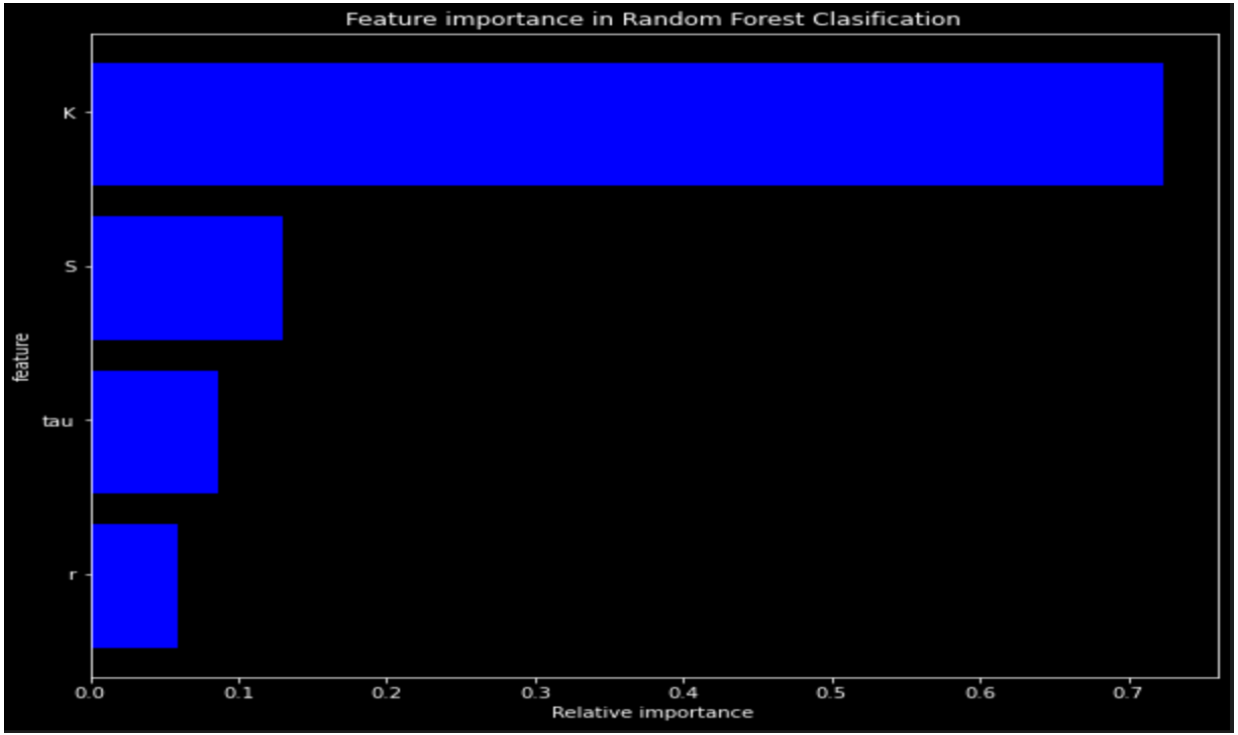
Random Forest Classification

- Supervised learning method using an ensemble of decision trees based on random samples
- The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees.
- Tuned the parameter using RandomizedSearchCV()

```
Classification error: 0.050746268656716390
```

N_estimators: 1000
Min_samples_split: 2
Min_samples_leaf = 1
Max_features = 'sqrt'
Max_depth = 20
Bootstrap = True

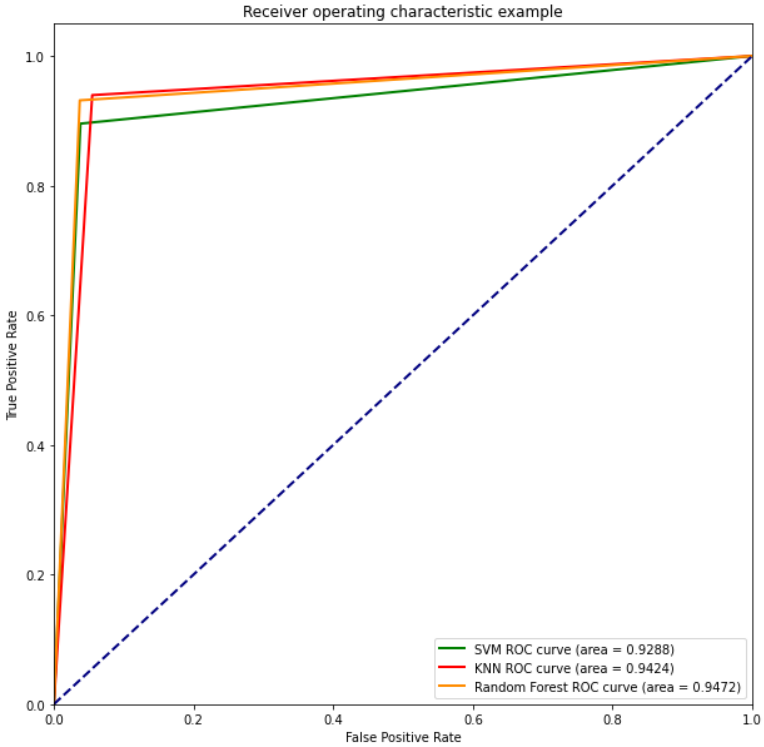
Random Forest Classification Feature Importance



Classification Model Comparisons

Method/Metric	Prediction Accuracy	Classification Error	ROC/AUC
Random Forest	**0.949**	**0.050**	**0.9472**
KNN	0.936	0.075	0.9424
SVM	0.935	0.065	0.9288

ROC/AUC Curve



1. Random Forest
2. KNN
3. SVM

Conclusion

Our group recommends the following methods

Regression: Random Forest/Decision Tree/SVM

(Very similar outcomes)

Classification: Random Forest

Business Understandings

- **In both prediction problems, would you argue if prediction accuracy or interpretation is more important? Why?**
  Prediction accuracy

- **Why do you think machine learning models might outperform Black-Scholes in terms of predicting option values?**

- Constant interest rate and volatility
- Normally Distributed underlying stock prices

Business Understandings

- **Can you argue from a business perspective that all four predictor variables should be included in your prediction (i.e., no variable selection is necessary)?**

  Huge potential for upside gains/downside losses in options trading

- **Are you comfortable about directly using your trained model to predict option values for Tesla stocks? Why?**

- Tesla stock trading around $1000
- Dataset contains stock prices under $450
- We do not know the company/industry related to the dataset