```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("uber.csv")
```

```
df.dtypes
```

Preprocess the data

```
df.isna().sum()
```

```
Unnamed: 0           0
key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     1
passenger_count      0
dtype: int64
```

```
df = df.drop(['Unnamed: 0', 'key'], axis=1)
```

```
df['dropoff_longitude'].fillna(df['dropoff_longitude'].mean())
df['dropoff_latitude'].fillna(df['dropoff_latitude'].mean())
```

Convert datatype of column 'pickup_datetime' from object to DateTime

```
df['day'] = pd.to_datetime(df['pickup_datetime']).dt.day
df['month'] = pd.to_datetime(df['pickup_datetime']).dt.month
df['year'] = pd.to_datetime(df['pickup_datetime']).dt.year
df['hour'] = pd.to_datetime(df['pickup_datetime']).dt.hour
df['dayofweek'] = pd.to_datetime(df['pickup_datetime']).dt.dayofweek
```

```
df = df.drop('pickup_datetime', axis=1)
df.dtypes
```

```
fare_amount          float64
pickup_longitude     float64
pickup_latitude      float64
dropoff_longitude    float64
dropoff_latitude     float64
passenger_count        int64
dtype: object
```

2. Identify outliers

```
df.plot(kind='box', subplots=True, layout=(7,2), figsize=(25,15))
```

```
df.iloc[:,0::]
```

```
def remove_outlier(df, col):
  Q1 = df[col].quantile(0.25)
  Q3 = df[col].quantile(0.75)
  IQR = Q3-Q1;
  lower = Q1 - 1.5*IQR
  upper = Q3 + 1.5*IQR
  df[col] = np.clip(df[col],lower,upper)
  return df

def treat_outliers_all(df, col_list):
  for c in col_list:
    df1 = remove_outlier(df, c)
  return df

df = treat_outliers_all(df, df.iloc[:,0::])
```

```
df.plot(kind='box', subplots=True, layout=(7,2), figsize=(25,15))
```

3. Check the Correlation

```
corr_matrix = df.corr()

plt.figure(figsize=(18,10))
sns.heatmap(corr_matrix,annot=True)
plt.show()
```

4. Implement linear regression and random forest regression models.

```
y = df['fare_amount']
x = df.drop('fare_amount', axis=1)
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2, random_state=30)
```

```
#Linear Regression Model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
```

```
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

```
from sklearn import metrics
r_squared_LR = model.score(x_train, y_train)
lr_RMSE = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
```